T · · Systems · · ·

rvsEVO

Version 6.01

User Manual

rvsEVO
Version 6.01
User Manual

Holzhauser Straße 4 - 8
13509 Berlin

Content

1 Introduction

In this chapter you will find a short description of rvs® and rvsEVO as well as an explanation of typographic conventions used in the present manual.

1.1 Short description of the system

What is rvs®

rvs® = Rechner-Verbund-System

The abbreviation rvs® stands for the German word **Rechner-Verbund-System**. The rvs® computer communication system is a well established base service for electronic data interchange (EDI).

rvs® serves to ensure transmission of electronic data between heterogeneous computer platforms using different network protocols.

To do so, rvs® implements a universal network model, which you can configure in each network node.

rvs® provides an efficient and reliable transport service for both standardized EDI message types and files of any format or contents. You can receive only such files that are explicitly destined for rvs®. This means that rvs® does not allow any unauthorized access to remote or to own data files.

The system was originally developed by Volkswagen AG and has been used in the German and European automobile industries for a number of years and by banks, insurances and industries worldwide.

rvs® uses the OFTP protocol.

What rvs® is not

rvs® is not an online system. It neither supports direct terminal-like access to other sites, nor provides a communication pipe from application to application on a data record level. You cannot directly execute transfers in your own application. You rather can place send orders from within your application to rvs® which will be handled asynchronously.

rvs® is not a job scheduling system.

rvs® does not care about the contents of the files it transfers. It only acts as a transparent transport medium and performs no semantic interpretation of the data it carries.

rvs® is not an EDI converter. Please contact your sales partner if you need to convert among specific message formats (e.g. VDA, ODETTE, EDIFACT, XML):

Phone: +49 375 606 19 902

E-Mail: rvs-service@t-systems.com

rvs® is not a network control or monitoring tool.

What is rvsEVO

rvsEVO is a communication software with a graphical user interface based, like rvs®, on the OFTP protocol.

Available features

The following features are available in rvsEVO 6.00:

- graphical user interface
- sending files to neighbour or to routed stations
- receiving files from neighbour or from routed stations
- receiving files on virtual stations; sending files from virtual stations
- activating neighbour stations
- support of the OFTP version 1.3, 1.4 and 2.0
- viewing information about receive, send, failed and ended jobs
- displaying monitor messages
- log files for tracing the monitor activities and for troubleshooting
- deleting or releasing EERPs (End-to-End-Response) if necessary
- archiving information about the send or receive jobs in the revision log
- code conversion with various code conversion tables
- format conversion
- defining job filters and actions when sending or receiving files
- compression and encryption
- backup and recovery
- No limit for size of transmitted files
- Support of Central Journal functions (for more information see the Central Journal User Manual)
- Support of SNMP Monitoring (for more information see the rvs® SNMP Agent User Manual)
- Remote access to rvsEVO-GUI (Remote GUI)
- Oracle database connection
- Microsoft SQL database connection
- PKI connection
- Multi-instance operation
- rvsEVO is implemented in Java. There exist four options of rvsEVO: rvsEVO Enterprise Edition, rvsEVO Standard Edition, rvsEVO Light Edition, rvsEVO Tiny Edition. Please see chapter 1.2 to 1.5 for description.
- Multi-instance capability

rvsEVO is implemented in Java. There exist four options of rvsEVO: rvsEVO Enterprise Edition, rvsEVO Standard Edition, rvsEVO Light

Edition, rvsEVO Tiny Edition. Please see chapter 1.2 to 1.5 for description.

rvsEVO uses a batch interface and file system to communicate with the application. If capable of doing so, the linked application can indicate successful processing and indicate successful dispatch.

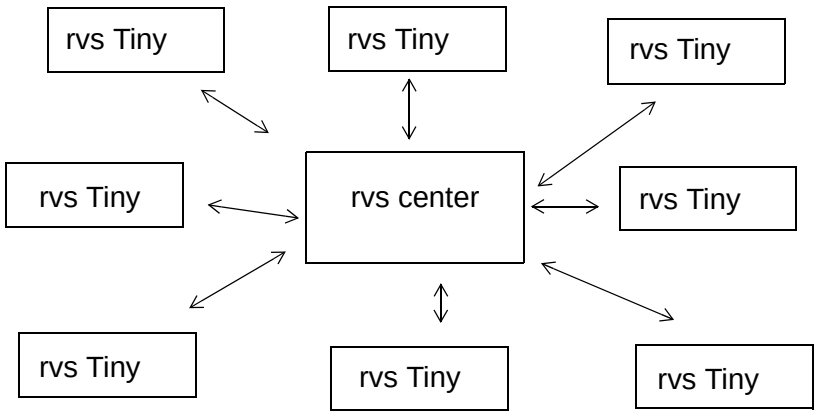
At present, rvsEVO supports the following networks: TCP/IP, TLS, ISDN (only for Windows), XOT, Proxy TCP/IP and Proxy TLS.

For more information on supported platforms please refer to the `$RVS_HOME/doku/readme.txt` release notes.

Note: Please read the chapter 1.6 for the explanation of `$RVS_HOME`.

1.2 rvsEVO Tiny Edition

The Tiny Edition supports the communication with only one neighbour station: rvs® center. The number of partner station is restricted to 1. Other stations can be reached via routing. Therefore rvsEVO Tiny Edition is particularly suitable for star topology.



Features of rvsEVO Tiny Edition (subject to modifications):

- 5 partner station (neighbour and routing stations)
- 1 neighbour station (rvs® center)
- 1 session
- max. 4 routing partner
- communication components TCP/IP and TLS
- encryption and compression
- Central Journal
- SNMP Agent

– code conversion (PC-Mainframe)

Additionally the following component can be linked:

– File Service Module

Hint: Please see the rvsEVO Product Datasheet for binding information of the technical data. You can download the datasheet from the following adress: `https://servicenet.t-systems.com/business-integration/rvs-produktdatenblatt/927428`.

1.3 rvsEVO Light Edition

Features of rvsEVO Light Edition (subject to modifications):

– max. 5 partner stations (direct or routing partner)

– 2 parallel connections (sessions)

– routing functionality

– communication components TCP/IP and TLS

– encryption and compression

– Derby database embedded

– Central Journal

– SNMP Agent

– code conversion (PC-Mainframe)

Additionally the following components can be linked:

– communication components ISDN (for Windows) and XOT

– File Service Module

Hint: Please see the rvsEVO Product Datasheet for binding information of the technical data. You can download the datasheet from the following adress: `https://servicenet.t-systems.com/business-integration/rvs-produktdatenblatt/927428`.

1.4 rvsEVO Standard Edition

Features of rvsEVO Standard Edition (subject to modifications):

– 999 partner stations (direct or routing partner)

– 20 parallel connections (sessions)

– routing functionality

– 1 virtual station

– communication components TCP/IP and TLS

– encryption and compression

– Derby database embedded

– Central Journal

– SNMP Agent
– code conversion (PC-Mainframe)
– remote GUI

Additionally the following components can be linked:

– communication components ISDN (for Windows) and XOT
– File Service Proxy
– further virtual stations

Hint: Please see the rvsEVO Product Datasheet for binding information of the technical data. You can download the datasheet from the following adress: `https://servicenet.t-systems.com/business-integration/rvs-produktdatenblatt/927428`.

1.5 rvsEVO Enterprise Edition

rvsEVO Enterprise Edition provides all the basic requirements for secure data transfer with a simultaneous performance even at high loads. rvsEVO Enterprise Edition includes an automatic adjustment of the processing capacity and uses a professional external database.

Features of rvsEVO Enterprise Edition (subject to modifications):

– 9.999 partner stations (direct or routing partner)
– 999 parallel connections (sessions)
– routing functionality
– 1 virtual station
– all communication components
– encryption and compression
– Oracle database connection
– Microsoft SQL database connection
– Central Journal
– SNMP Agent
– code conversion (PC-Mainframe)
– PKI connection
– File Service Proxy
– remote GUI (10 users)
– Multi-instance operation

Additionally the following components can be linked:

– rvs® OFTP Proxy for OFTP2
– further virtual stations

Hint: Please see the rvsEVO Product Datasheet for binding information of the technical data. You can download the datasheet from the following adress: `https://servicenet.t-systems.com/business-integration/rvs-produktdatenblatt/927428`.

The available features are limited by the license key.

For more Information on rvsEVO Editions please contact your sales partner (Phone: +49 375 606 19 902; E-Mail: rvs-service@t-systems.com)

1.6 Representation means

This chapter describes the typographic conventions used in this manual and explains the meaning of specially highlighted expressions.

Typographic conventions

- Instructions begin with a bullet.

– Other lists begin with the en dash.

Character styles

`Courier`	Commands, menu commands, file names, path names, programs, examples, scripts, options, qualifiers, data sets, fields, modes, window names, dialog boxes and statuses
BOLD and **UPPER CASE**	Parameters, environment variables, variables
"Inverted comma"	Links to other manuals, sections and chapters, literature
Bold	Important terms, names of operating systems, proper names, buttons, function keys

Directories:

$RVS_HOME As user directories are found on different locations for the different operating systems we use the variable **$RVS_HOME** in this manual. Default values are:

– `C:\Programs\rvsEVO` for **Windows**.

Substitute the variable with your correct path.

1.7 Target group

This manual is meant for regular users of rvsEVO as well as administrators. It provides an overview of the basic rvsEVO functions.

Skills The following skills are required to be able to use rvs EVO:

– good knowledge of the current operating system

– knowledge of the communication techniques in use TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP oder Proxy TLS.

Before starting to work with rvsEVO it is advisable to have read this book.

2 Installation

This chapter describes the system requirements as well as the rvsEVO installation procedure.

2.1 System Requirement

To successfully operate rvsEVO you need the following software:

Software

– Operating system: Windows XP / Windows 7 / Windows 8.1 / Windows Server 2008 R2 / Windows Server 2012 R2 or UNIX (AIX, Solaris/ SunOS, HPUX, Linux, zLinux).
– Java runtime environment Version (JRE) 1.7_XX or higher.

Hint: With Windows and Linux systems the Java runtime environment Version 1.8.0_05 is installed automatically during the installation. With UNIX systems you can select an installed JRE during the installation.

Initially, you need at least 400 MB free space on your hard disk. Depending on the amount of usage, the retention period for old entries, and the time between database cleanups, the space requirement may be considerably larger.

Attention: Please note that there are different installer for 32-bit and 64-bit operating systems for Windows and Linux systems. rvsEVO will not work properly when using an installer for 64-bit systems on 32-bit operating system. The other way round you can use an installer for 32-bit systems on 64-bit operating systems.

Auf einem 32 Bit-Betriebssystem, können Sie nicht den Installer für ein 64-Bit-Betriebssystem verwenden. Anders herum ist auf einem 64-Bit-Betriebssystems der Installer für ein 32-Bit Betriebssystem verwendbar

You can download rvsEVO from the following web page:

https://servicenet.t-systems.com/business-integration/software/958408

If that is not possible for you, please contact your sales partner:

Phone from Germany: 0800 664 77 45;
Phone from other countries: +49 375 606 19 902
Email: rvs-service@t-systems.com

We can send the software also on DVD to you.

Networks

Communication line based on TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP und Proxy TLS.

Hint: ISDN is only available with Windows32 installer.

ISDN system requirement

ISDN is only for Windows platform available. For Unix platforms XOT should be used.

If you want to use rvsEVO to exchange data via the **ISDN network** you need the following equipment:

– ISDN type telephone connection with s_0 bus having at least two data channels (B-channel) and one control channel (D-channel)

– ISDN adapter

– CAPI 2.0 driver software for operation of the ISDN card under Windows.

How to install the ISDN interface:

- Install the ISDN card into your computer and attach the ISDN card to the ISDN connection.
- Make sure that the ISDN card works properly.

Note: Many card manufacturers supply suitable software for a self-test, such as a call from one data channel to another. Restart your computer after installation and the test in order to make the CAPI 2.0 driver available for other applications.

External ISDN router

If you use an external ISDN router with remote CAPI Interface (e.g. BinTec Brick), you do not need an internal ISDN card. In this configuration, several applications can share the same Brick router. The Brick router supports the "Remote CAPI" interface. This means that every computer in your LAN uses the Brick router as if it were a local ISDN card in the computer.

Note: T-Systems has successfully tested the following ISDN devices for use with rvsEVO:

ISDN Devices

Device	Manufacturer
EICON DIVA Server BRI-2M	Eicon, http://www.eicon.de
EICON DIVA Pro 2.02 PCI	Eicon, http://www.eicon.de
EICON DIVA 2.0	Eicon, http://www.eicon.de
Longshine LCS-8051A	Longshine, http://www.longshine.de
BIANCA/BRICK-XS, -XM, -XL2; X4000, X4100, X4300, R1200, R4100 und R4300	Funkwerk EC, http://www.funkwerk-ec.com

rvs® Service Support Center will provide you with the current list of tested devices:

Phone from Germany: 0800 664 77 45;
Phone from other countries: +49 375 606 19 902
Email: rvs-service@t-systems.com.

XOT systems requirement

On demand we shall send you a separate document with examples how to configure CISCO or BinTec router for XOT.

To use the XOT functionality in rvsEVO you need a TCP/IP connection to the XOT-capable router e.g. CISCO 801, CISCO 2600 or BINTEC X4300, R1200, R4100 and R4300. (CISCO: http://www.cisco.de; BINTEC: http://www.funkwerk-ec.com).

2.2 Obtaining a License

You need a license key to work with rvsEVO.

rvs® after sales service

Please contact the rvs® Service Support Center to receive a license key

phone from Germany: 0800 664 77 45
phone from other countries: +49 375 606 19 902
email: rvs-service@t-systems.com.

To receive a license key:

– Please send the hostname and the **Odette** ID to the rvs® Service Support Center. You get the hostname as typing hostname in the command prompt window (`Run -> cmd`). You can request the Odette ID with the VAD (`http://www.vda.de/de/verband/fachabteilungen/logistik/infos/odette-id/index.html`)
– You will be sent your licensekey file by email.
– Save the license key file in the `$RVS_HOME\conf\` directory as license.properties.

Note: For more information about the ODETTE-ID see chapter 5.2.2 "Parameter Groups of Local Station" .

Renew a License

Use the tool `importLicense` to import a new license key at runtime. This could be required if your license key expires or if you need a license for added rvsEVO components. Please see chapter 19.6.15 "importLicense: Import of rvsEVO License Key" for more information.

Test License

For receiving a test license, please contact the rvs® Service Support Center too. In this case you must not inform the rvs® Service Support Center about your hostname and the Odette ID.

2.3 New Installation of rvsEVO

This chapter describes the installation of rvsEVO. Please read the chapter 2.1 "System Requirement" before installing.

Attention:

For successful installation the installing user needs execute rights for `TEMP` directory (defined via **TEMP** environment variable). If he has no execute rights he needs to create a `TEMP` directory with execute rights

before the installation procedure is started and to define this directory via **$IATEMPDIR** environment variable.

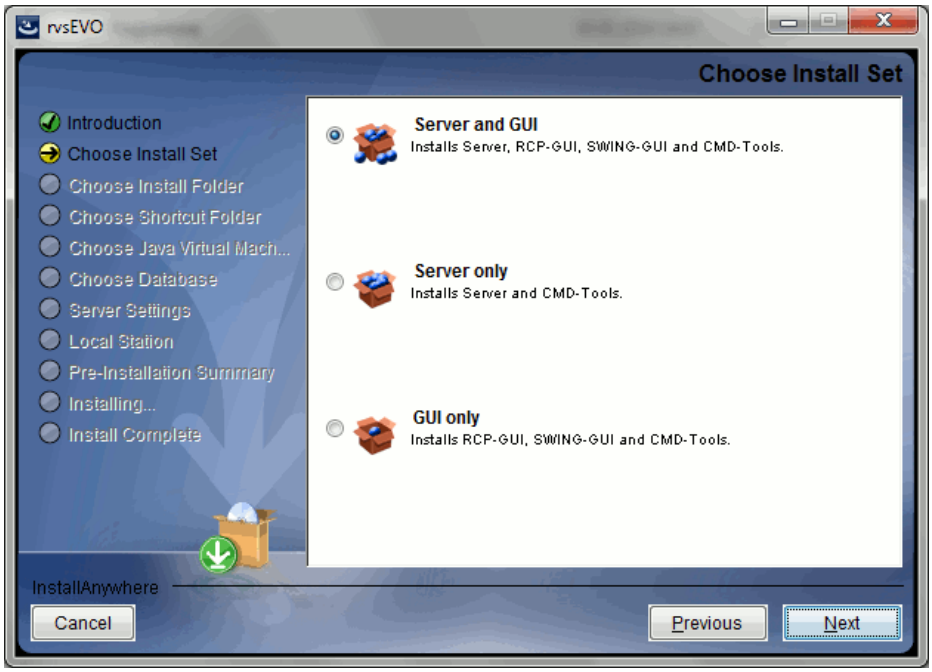
If you want to install rvsEVO Enterprise Edition and use an Oracle database from version 12 on, please make the following Oracle setting **before installing of rvsEVO**: Assign the quota „unlimited“ to the Oracle database user for the table space used by him. Failure to follow this steps will cause the installation to fail.

Hint: rvsEVO can be installed without input of any data. You only have to press the return button. In this case the configuration will done with the default values. Exception: this procedure cannot be executed in german language on console.

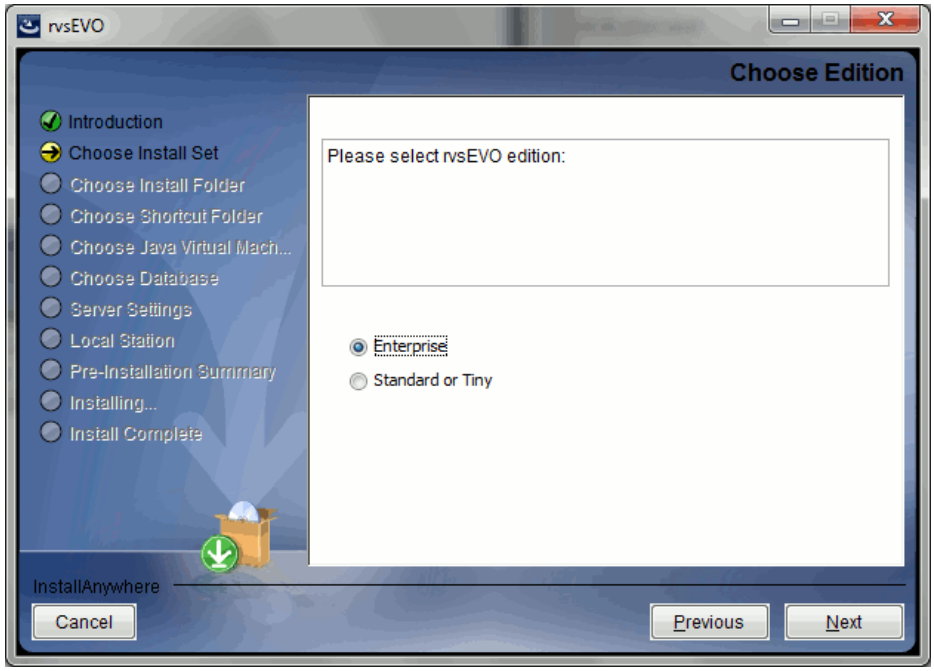
2.3.1 Installation on Windows Systems

- Start Windows and log in as a Windows user with administrator rights. **Hint:** Unlock the firewall before the installation procedure for the ports of the listeners that should be used for receiving data from your partner station (if the Windows firewall is enabled).
- Start the installer `rvsEVO_setup_Windows32.exe` respectively `rvsEVO_setup_Windows64.exe` (for 32 bit or 64 bit OS) by double-clicking or using the Windows command: `Start -> Run`.
- With the first window you get information on installation of rvsEVO.
- Thereafter you can choose the language of the installation wizard.
- If there is already an rvsEVO installation on your machine, you have to decide whether you would like to install a new version or to update your current rvsEVO. If you decide to upgrade your existing rvsEVO please continue how descript in chapter 2.4 "rvsEVO update installation", else click on **Install** button for going on.

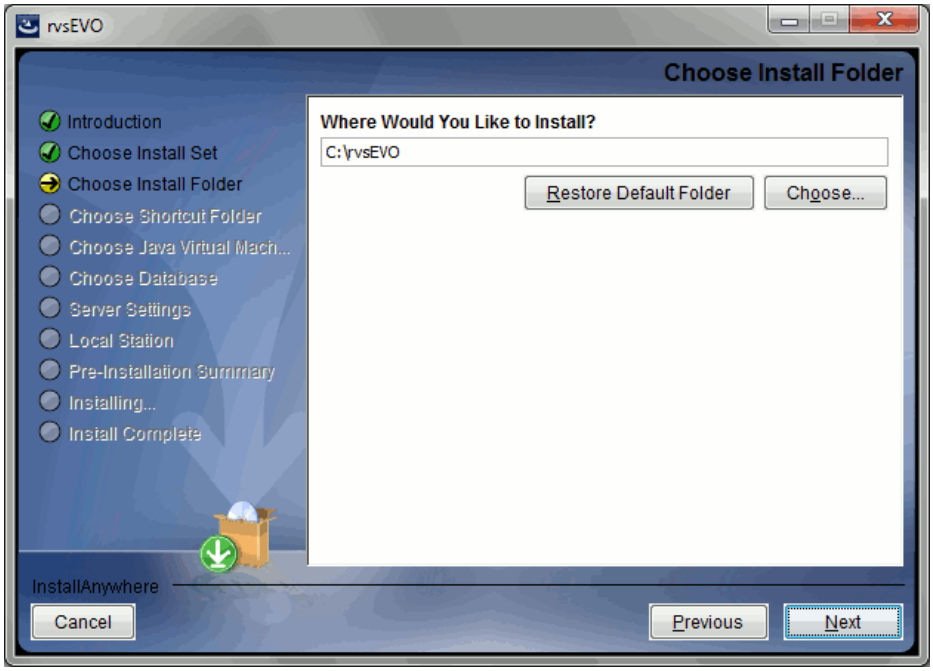
- Next you have to choose the install set. Information about Client-Installation (only GUI) you can find in chapter 4.3 "Remote GUI"



- In the following window you can decide which rvsEVO edtion is to be installed.



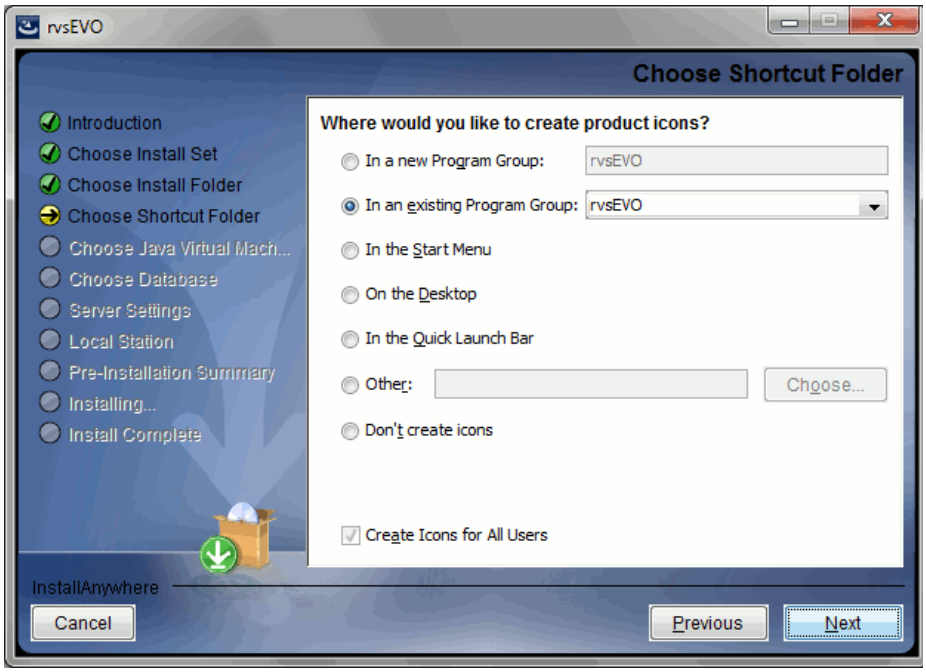
- In the dialog below you can indicate the rvsEVO destination directory. This directory may not exist.



If you choose an existing directory you are asked to choose another directory name.

- Define the program group to install rvsEVO icons in the dialog below.

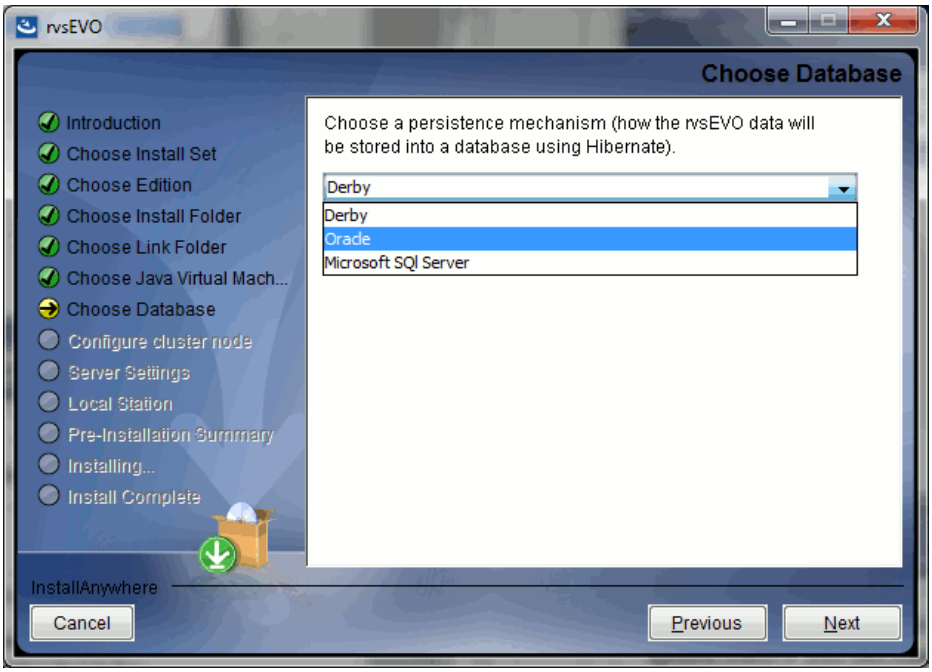
  Hint: The Checkbox **Create Icons for All Users** must be enabled if this setting should be valid for all users.



The following dialogs refer to installation of Enterprise Edition only

- With installation of rvsEVO Enterprise Edition you can choose the database in which your data should be saved. Oracle and MS SQL database are available.

  Important: You cannot run rvsEVO Enterprise Edition with Derby database.



- Set your Oracle connection parameters if you decided to use an Oracle database:
  - **jdbc.url** has the following syntax:

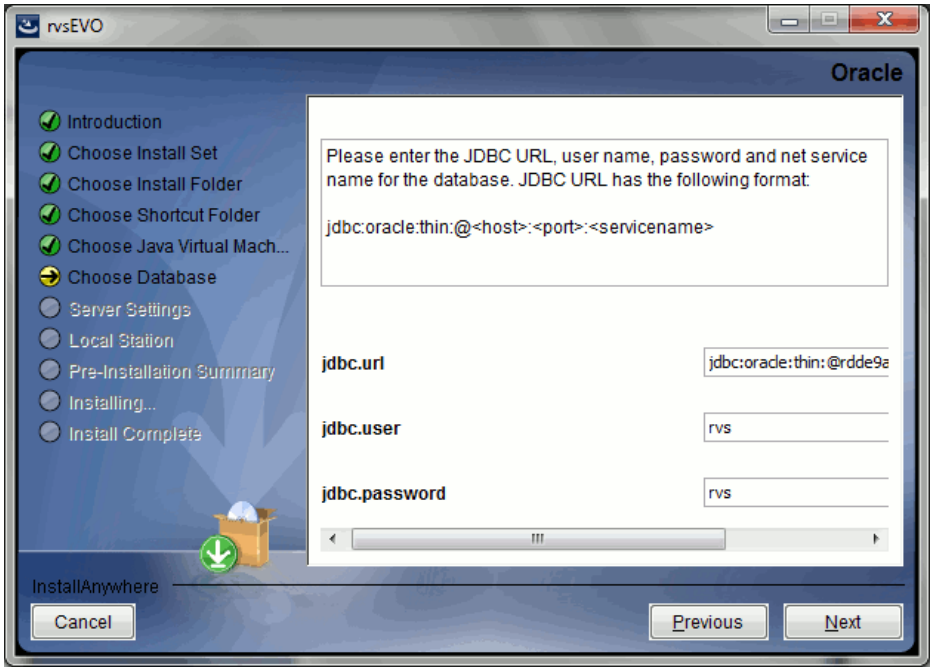
    `jdbc:oracle:thin:@<server>:<port>/<service_name>`

    `server` is the name or IP address of the machine where Oracle server is installed.

    Default port for Oracle is 1521.

    `service_name` is Oracle service name.
  - **jdbc.user** is the user setting up on Oracle database
  - **jdbc.passsword** is his password.

    For further information see chapter 19.3 "Oracle"

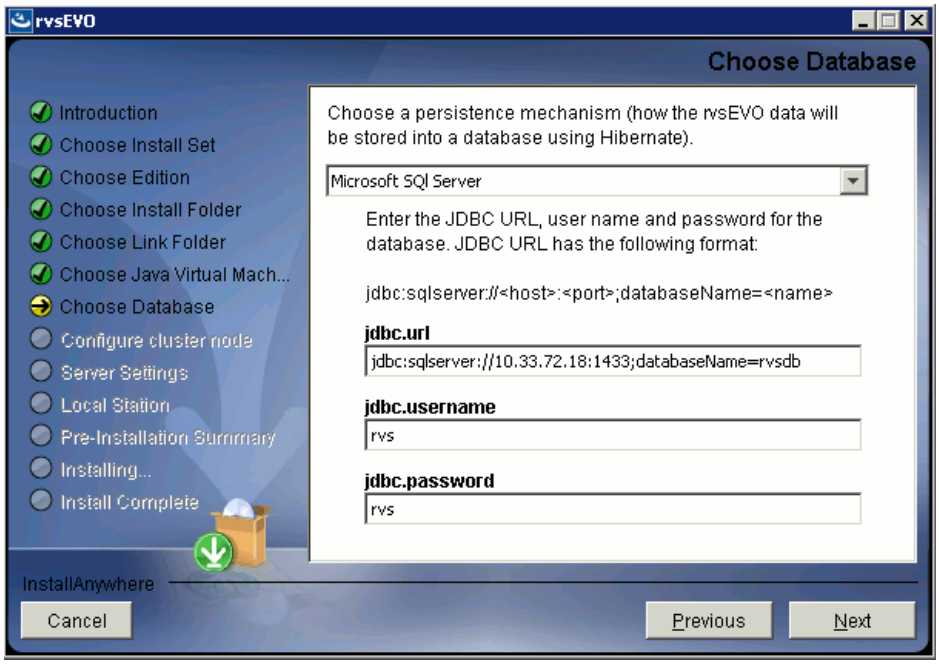


Analogous to Oracle database, set the MS SQL connection parameters with MS SQL database:

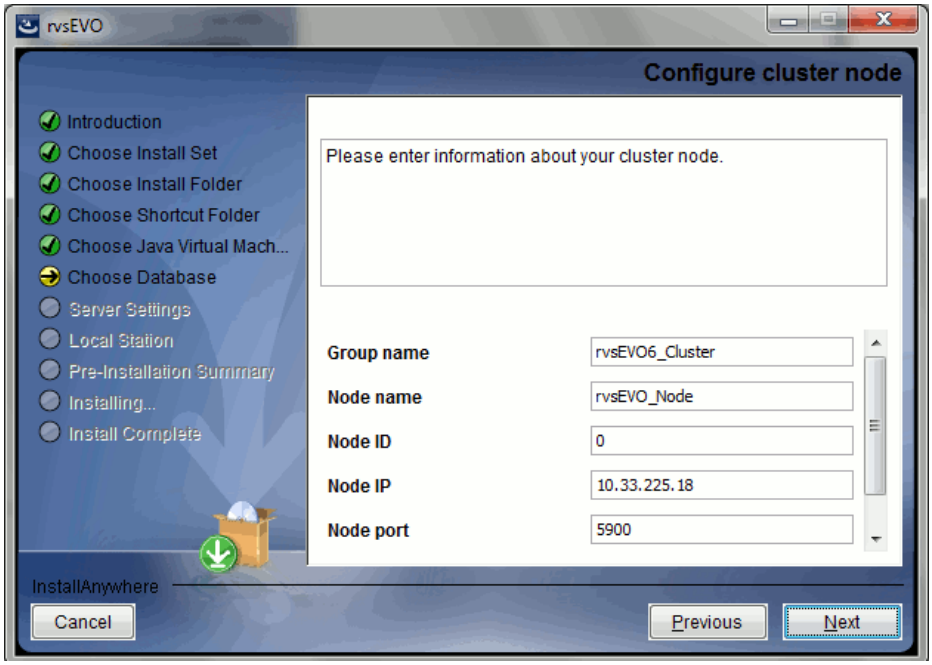
– **jdbc.url** has the following syntax:

  `jdbc:sqlserver://<host>:<port>;databaseName=<name>`

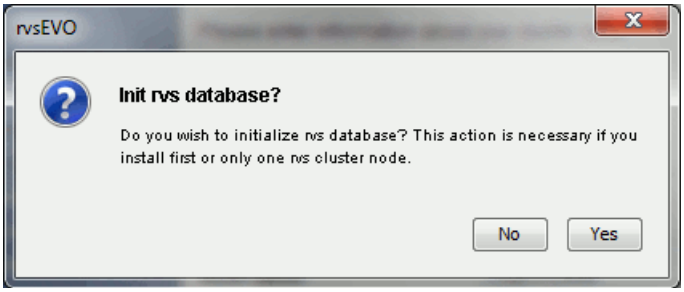
  Default port for MS SQL is 1433.



- The following window contains the parameters for setting up the cluster node. **Group name** is the name of your server farm. The Group name must be the same on all cluster nodes. **Node name** is an unambiguous designation of the node to be installed. Furthermore you were asked for the **Node IP,** the **Node port** and the IP addresses or hostnames of the other node of the server farm (**Nodes**).



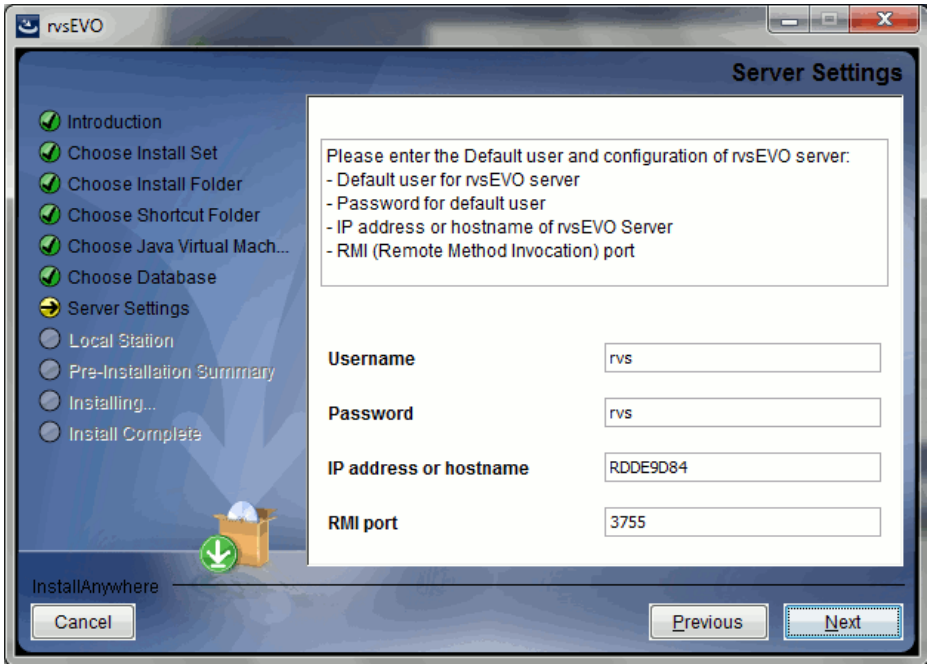
- The dialog below offers to initialize the database. If installing a server farm you have to initialize the data base only during installing the first node. Please see chapter 15 "Multi Instance Functionality of rvsEVO" for more information about cluster functionality..



Continuance for all editions

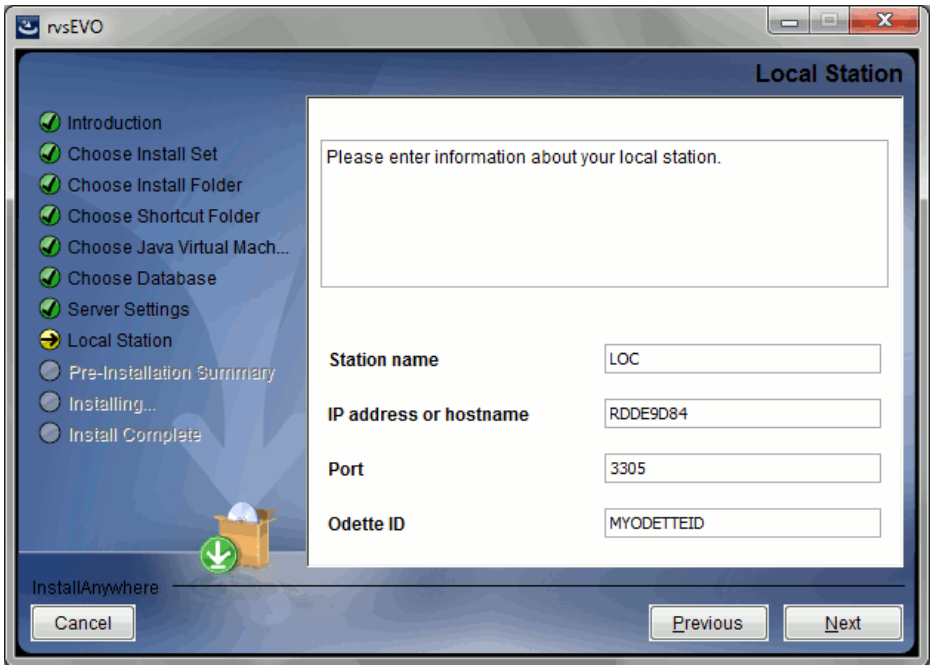
- Subsequently please define the username and the password of the default user with administrator rights as well as the IP address or hostname (default: hostname of your machine) and the RMI port (default: 3755).

  Hint: Due to the special role of the default user concerning the local Client Server communication you cannot change his data in the User Management.



- If you have already installed rvs® portable on your system, you can define to import the settings from rvs® portable. In case of migration from rvs® portable click the **Yes** button. Please read chapter 2.5 "Migration from rvs® portable to rvsEVO" for further information about migration from rvs® portable.
- If you install just the GUI you can also ignore the dialog below. It deals with the settings of the local station. Please read the chapter 5.3.3 "Parameter Groups of the Partner Station:" for more information about setting the station parameters (e.g. how to obtain the ODETTE ID). The neighbour station will be configured after installation

procedure via GUI. (See chapter 5.3.1 "Add a Neighbour Station").



- In the next dialog you are given a brief overview of selections you have made (installation directory, link directory). The required and the currently available disk space is also displayed. Press the **Next** button to start installation and to copy the installation files into the directories you specified.
- The last dialogs informs you about the successfully installation of rvsEVO.

2.3.2 Installation on Linux and UNIX Systems

Installation on UNIX systems runs nearly analog to an installation on Windows systems. The installation file for Generic UNIX is named `rvsEVO_setup_GenericUnix.bin`. For Linux systems please use `rvsEVO_setup_Linux32.bin` for 32 bit OS and `rvsEVO_setup_Linux64.bin` for 64 bit OS.

Hint: There is the following difference on UNIX systems:

Java runtime environment

There is an additional dialog for defining the JRE (Java runtime environment) for rvsEVO operation. The installer searches for installed components on your systems and proposes the versions found in a dialog. rvsEVO has been released for use of JVM version 1.7._X onwards.

For encryption rvsEVO uses an extension of Java runtime environment - JCE (**J**ava **C**ryptography **E**xtension) of Oracle Corporation to implement the cryptographic features. Due to import control restrictions of some countries, the JCE jurisdiction policy files allow "strong" but limited cryptography to be used. It is necessary to install these extension of the Java runtime environment to use unlimited cryptographic strengths. This is available for most countries.

For more information about JCE files read the `$RVS_HOME/docu/readme.txt` file.

Important:

Without JCE you cannot transfer encrypted files. If you try to transfer encrypted files and you did not install JCE, you will get the following message: „invalid key length“.

If you use a central JRE: After updating your operating system the JCE settings could be lost. In this case it is strictly necessary to install the JCE again.

Installation Modes

The installation on UNIX and Linux systems can be done in different modes: `console` (default), `awt,` or `silent`.

For graphical modes awt the UNIX environment variable **DISPLAY** for the X-Server should be set.

Example:

```
export DISPLAY=<IP address of X Server>:0.0
```

Note: Make sure to call the installation file as a shell script when you perform installation. Default mode is the console mode.

Example:

```
sh ./rvsEVO_601_00_SE_setup.sh
```

The installation prompts are identical in both modes (see section 2.3.1 "Installation on Windows Systems").

Hint: There are minor differences for particular UNIX platforms mentioned in the release notes for the respective version (`$RVS_HOME\doku\readme.txt` document).

2.4 rvsEVO update installation

Requirements

The system variable **RVS_HOME** is to be set for the logged on user (the value of **RVS_HOME** is the installation directory of rvsEVO). Set **RVS_HOME** by choosing `Start -> Control Panel -> System -> Advanced -> Environment Variables`.

- Start the `rvsEVO-setup-X.XX.XX.exe` installer by double-clicking or using the Windows command: `Start -> Run`.

  Attention:

  Use a Windows32 installer if you communicate via ISDN. On the other hand an error message is displayed.

- Click the **Update** button if the installer asks you whether you would like to install a new version or to update your current rvsEVO.

- In the next dialog you are given a brief overview of selections you have made (installation directory, link directory). The required and the currently available disk space is also displayed. Press the **Next** button to start installation.
- The last window informs you about the successfully installation of rvsEVO. Finish the installation routine by pressing **Done** button.
- Use the tool `$RVS_HOME/tools/importKeystores` for importing the certificates from old keystore files into the rvsEVO database (see chapter 19.6.14 "importKeystores: Import of a Keystore File").

Important: During the update installation a new `$RVS_HOME/conf/rvs-system.properties.new` file is added. Please take over the configurations of your old `rvs-system.properties` file to the new one and rename `rvs-system.properties.new` in `rvs-system.properties`.

In case of an update installation from rvsEVO Release 5.XX to 6.XX the certificate usage has to set up again. This configuration can be done via station configuration -> parameter group Keys.

2.5 Migration from rvs® portable to rvsEVO

Requirements

For migration of settings from rvs® portable you have to consider the following requirements:

– an executable rvs® portable must be installed on your system

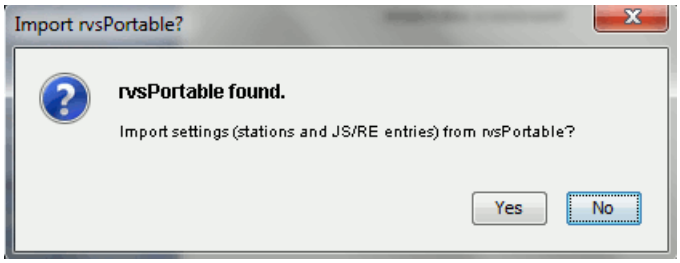
– migration should only be started if the absence of communication has been assured (no file reception/transmission and no encryption/ compression)

– The system variable **RVSENV** is to be set for logged on user. Set **RVSENV** by choosing `Start -> Control Panel -> System -> Advanced -> Environment Variables`.

The following steps need to be done for migration from rvs® portable to rvsEVO:

- Install rvsEVO on the same machine where rvsXP / rvsX is installed like described in chapter 2.3.

- The installer searches through your system for an existing rvs® portable. If you are asked whether you wish to import the settings from rvs® portable, press the **Yes** button.



- Only with encrypted transmission: import your own key pair with the program `importComSecureKeyPair` (see chapter 12.10.8 "importComSecureKeyPair").

  Important: If the public key is not existent in format X.509 you have to insert the additional information in the file `$RVS_HOME\conf\certificate-properties.xml`.

  Sample file `(certificate-properties.xml):`

```
- <!--  common-name  -->
  <entry key="subject.cn">rvsEVO-comsecure-j</
entry>

- <!--  organisastion-unit  -->
  <entry key="subject.ou">rvs</entry>

- <!--  organisastion  -->
  <entry key="subject.o">T-Systems International
GmbH</entry>

- <!--  locality  -->
  <entry key="subject.l">Berlin</entry>

- <!--  state  -->
  <entry key="subject.st">Berlin</entry>

- <!--  country-code  -->
  <entry key="subject.c">de</entry>

- <!--  email  -->
  <entry key="subject.email">rvs-support@t-
systems.com</entry>
```

  The following values are set in above example:

  – `common name` = rvsEVO-comsecure-j
  – `organisation unit` = rvs
  – `organisation` = T-Systems International GmbH
  – `locality` = Berlin
  – `state` = Berlin

  - `contry code` = de
  - `email` = rvs-support@r-systems.com

  The other elements of `certificate-properties.xml` must not be edited.

- Only with encrypted transmission: import the public keys of your partners.
- Create the users which were defined in rvs® portable. They will not be created automatically.

Note: Use the following command tools for importing the settings into an existing rvsEVO:
`$RVS_HOME/bin/portable2codetable.bat`
`$RVS_HOME/bin/portable2jobstart.bat and`
`$RVS_HOME/bin/portable2stationlist.bat`

portable2codetable

Use this tool for converting a codetable from rvs® portable (rtc format) to xml format.

Syntax:

`portable2codetable <rtc file> <xml file>`

`rtc file` is the name of the codetable in rvs® portable. `xml file` is the name of the codetable in rvsEVO.

portable2jobstart

Use this tool for creating a jobstart list (xml file) from rvs® portable. rvs® portable must be launched.

Syntax:

`portable2jobstart <file>`

`file` is the xml file name. The file can be imported into the rvsEVO database via `$RVS_HOME/tools/importJobstarters` script.

portable2stationlist

Use this tool for creating a stationlist (xml file) from rvs® portable. rvs® portable must be launched.

Syntax:

`portable2stationlist <file>`

`file` is the xml file name. The file can be imported into the `rvsEVO` database via `$RVS_HOME/tools/importStationlist` script.

2.6 rvsEVO Version

Via the `Help` functionality in the menu bar you get information about the installed rvsEVO version. As an alternative you can execute the script `$RVS_HOME/tools/rvsver.bat`.

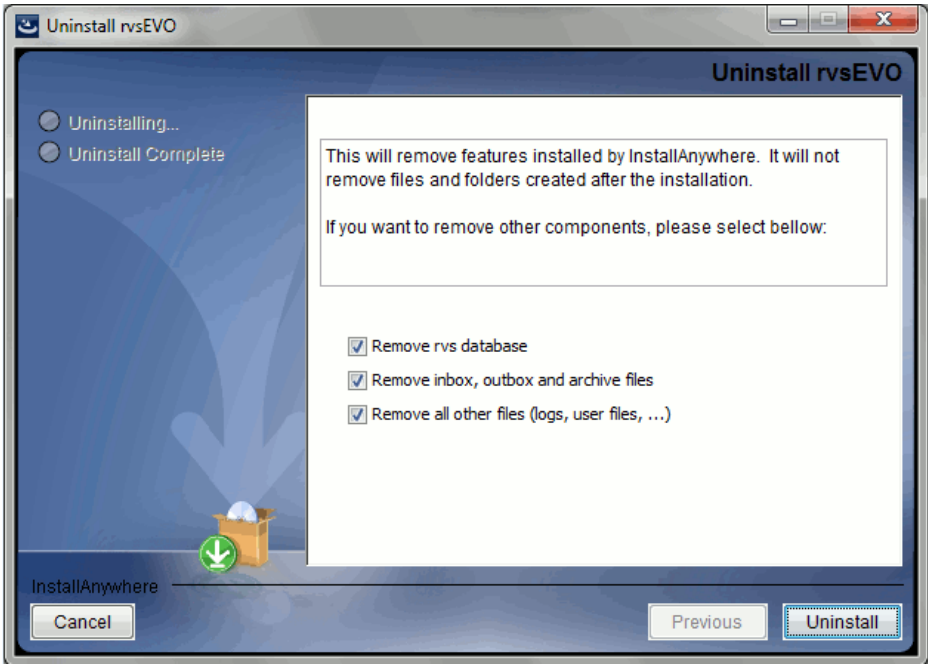
2.7 Uninstall rvsEVO

In order to uninstall rvsEVO, please follow the procedure given below:

Windows:

- Start the program `$RVS_HOME\UninstallerData\Uninstall.exe` and follow the dialog.
- In the first dialog you can indicate the components to be deleted. Those are available:
  - `Remove rvs database`
  - `Remove inbox, outbox and archive files`
  - `Remove all other files (logs, user files ...)`

  Click on **Uninstall** button to uninstall rvsEVO.



- The next window displays the components which have not been deleted. Click on **<done>** in order to exit the uninstalling program.

Unix

use the shell-script `$RVS_HOME/UninstallerData/Uninstall rvsEVO_StandardEdition.sh`.

Uninstall procedure on UNIX-systems can be executed analogous to rvsEVO installation. Following modes are available: `awt, silent` and `console`. For the graphical mode `awt`, the system environment variable **DISPLAY** must be set for the X server (see example in chapter 2.3.2). The uninstall routine is executed in the console mode by default. The uninstall queries are the same in all modes.

3 Start and Stop of rvsEVO

3.1 Starting of rvsEVO Server

The server is to be started before the graphical user interface can be started.

Windows: Start the rvsEVO Server either via the program group: `Start -> All Programs -> rvsEVO -> Server` or via command line: `$RVS_HOME\bin\startServer.bat`.

Unix: Start the shell script `$RVS_HOME/bin/startServer.sh`.

Hint: You get the following error messages if you try to boot an already started rvsEVO server:

– java.sql.SQLException: Failed to start database 'C:/rvsEVO/db' with class loader sun.misc.Launcher...

– java.sql.SQLException: Another instance of Derby may have already booted the database C:\rvsEVO\db.

3.2 Start GUI

Language setting

The default language of the GUI is English. The language setting of the GUI can be aligned via `FirstLanguage` parameter `$RVS_HOME\conf\rvsEvoClient.prefs` file (`en`=English, `de`= German). This setting does not concern the log messages. Changes only take effect after rvsEVO GUI restart

Example for language setting English:

```
<entry key="client.FirstLanguage" value="en"/>
```

Start the GUI via the program `$RVS_HOME\bin\startGUI`.

Syntax:

```
startGUI [-rcp -swing -local -console -reset -help -?]
```

Optional startGUI Parameters:

Parameter	Description
-rcp	RCP GUI is started on local machine (default on Windows and Linux systems; not available on UNIX systems).
-swing	SWING GUI is started on local machine (default on UNIX systems).
-local	Server and GUI are started in the same process.
-console	GUI is started with console output.
-reset	GUI display settings are resetted during startup

Optional startGUI Parameters:

Parameter	Description
-help	Requests help information (usage).
-?	Requests help information (usage).

Hint for Windows:

The GUI can also be started via rvsEVO program group (`Start -> All programs -> rvsEVO -> rvs GUI`).

3.3 rvsEVO as Windows service

By default, rvsEVO is installed as a Windows service.

Note: The term service means a program that can be started from the operating system and works in the background.

rvsEVO as Windows service can also be started via the batch script `rvsEVOservice.exe`.

Syntax:
`rvsEVOService <-install [ini]> <-configure [ini]> <-remove [ini]> <-debug [ini]> <-run [ini]>`

Optional rvsEVO Service Parameters:

Parameter	Description
-install [ini]	Installs rvsEVO as service. `$RVS_HOME\bin\rvsEVOService.ini` is used by default.
-configure [ini]	For new configuration of an installed rvsEVO Windows service. `$RVS_HOME\bin\rvsEVOService.ini` is used by default.
-remove [ini]	Removes rvsEVO as service. `$RVS_HOME\bin\rvsEVOService.ini` is used by default.
-debug [ini]	For starting a console application for debugging. `$RVS_HOME\bin\rvsEVOService.ini` is used by default.
-run [ini]	Starts rvsEVO as service on console. `$RVS_HOME\bin\rvsEVOService.ini` is used by default.

Hint: After installation of rvsEVO as a service, you can find rvsEVO in the list of system services (`Start - > Control Panel -> Adminis-`

trative Tools -> Services). If you want to start rvsEVO with each system start, you can set the startup type to Automatic by choosing the Automatic entry from the combo box in the **Startup type** area.

3.4 Automatic Start on UNIX Systems

By default, the shell script $RVS_HOME/bin/startServer.sh is used for starting the rvsEVO server.

The $RVS_HOME/tools/scripts/Linux and $RVS_HOME/tools/scripts/Solaris directories contain scripts that enable the automatic start and stop of rvsEVO together with the operation system. In the same directories you can find text files, in which the installation steps are described.

3.5 Stopping of rvsEVO server

Windows: Stop rvsEVO by choosing File -> stop Server in the menu bar of rvsEVO GUI.

As an alternative you can also stop rvsEVO at the command prompt by the $RVS_HOME\bin\stopServer.bat program.

Unix: Stop rvsEVO by starting the shell script $RVS_HOME/bin/stop-Server.sh.

4 Graphical User Interface

4.1 Introduction

By using the GUI (**G**raphical **U**ser **I**nterface) you don't need a lot of practise for working with rvsEVO. The most important functionalities can be executed via the GUI.

From Version 6.00 upwards rvsEVO provides an upgraded user interface which can be adjusted to your requirements.

Perspective

A Perspective is the active workspace in rvsEVO. The following perspectives are available: `Transfer, Stations, Key Management, Automation, Parameter, User Management` and `Monitoring`.

View

Each workspace can be represented in different views (e.g. tree view, list view, detailed view). You can re-position a view via **Drag & Drop**. Thereby, a frame gives you a preview of the new position and the docking site is displayed by an arrow.

Via a right-click on the tab a dropdown list opens and the following functionalities are avaiable:

– `Close` (the active view)

– `Close others` (the views in background)

– `Close all.`

Hints:

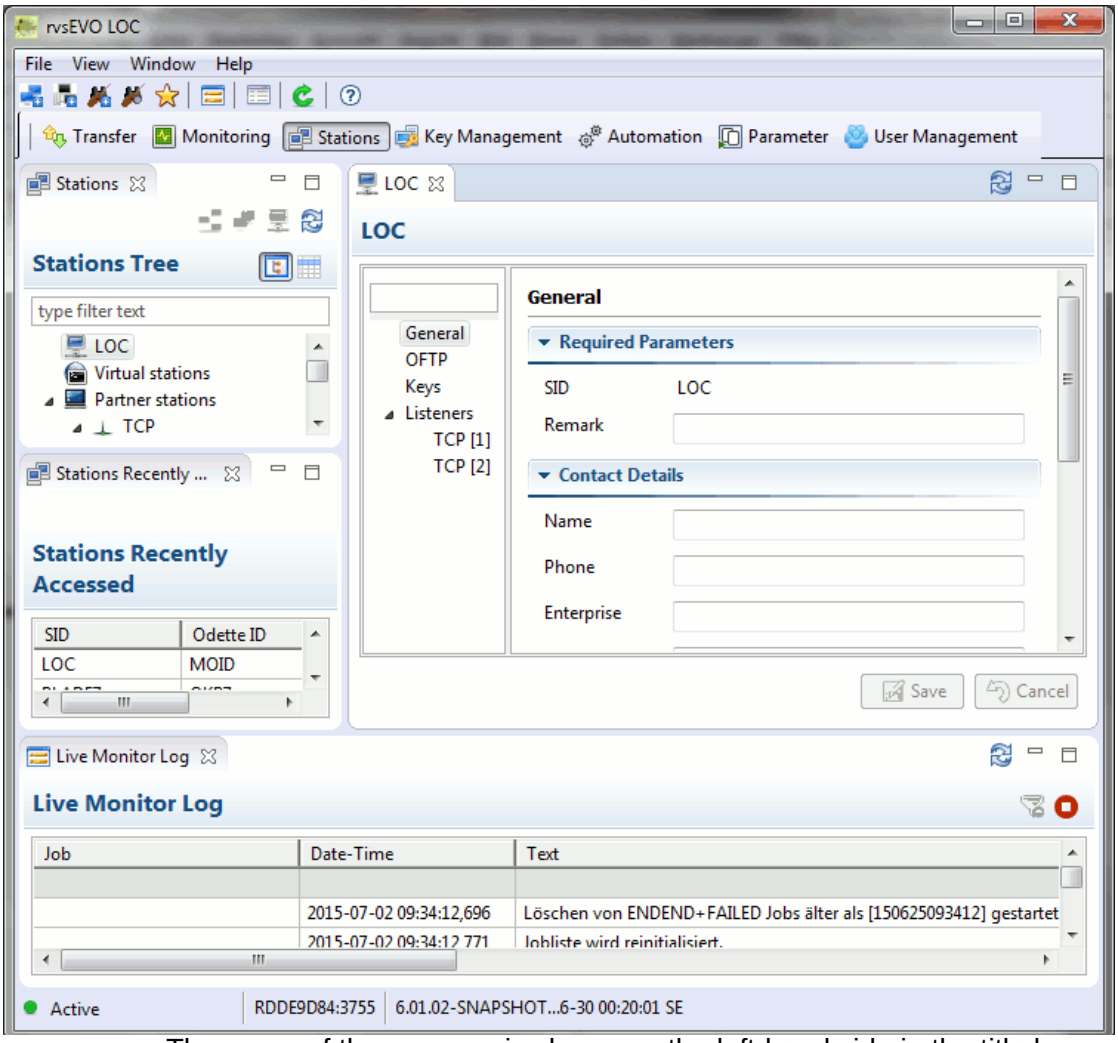
Minimized windows can be restored with a mouse click on the symbol at the top of the window.

Drag the GUI window to the top of the screen to maximize the window. Draw the GUI window to the right or left edge of the screen to change to half-screen. Draw the window to the center of the screen to go back to previous position.

A mouse click on the menu item `Window -> Reset Perspective` resets the perspective to default view.

4.2 Local GUI

The illustration below shows the basic structure of the GUI in the Standard Edition of the Stations perspective.



The name of the program is shown on the left-hand side in the title bar and on the right-hand side you see the minimize, maximize and close symbols.

Thereunder you can find the menu bar which offers the following functionalities:

– `File`: Exit (Remote GUI: Logout, Exit)

– `View`: Selection of perspective (Transfer, Stations, Key Management, Automation, Parameter, User Management, Monitoring)

– `Window`: Reset Perspective, Close Perspective, Open Perspective, Show View)

– `Help`: rvsEVO help

With a click on one of the right-hand buttons in the function bar you can select the perspective. The symbols on the left side are described elsewhere.

The middle part is represented in different views. In default view on left-hand the navigation area is arranged and on right-hand you can find the the work area.

At the bottom of the GUI the log messages are displayed (Please see chapter 7.2 "Live Monitor Log" for further information). Thereunder the status line is arranged and gives information about rvsEVO installation.

4.3 Remote GUI

In rvsEVO you can login via remote access.

Pre-requisites

The following prerequisites should be fulfilled for a remote connection:

- An rvsEVO client should be installed on the client-side (see chapter 2.3 "New Installation of rvsEVO").
- The client user is to be set up as a user, on the server side.
- The server should be started

4.3.1 Starting the GUI from remote computer

You can start rvsEVO GUI via the rvsEVO-program group: Start -> `All programs -> rvsEVO -> GUI.`

Alternatively, this function can be started in the command line, with the script `$RVS_HOME/bin/startGUI.`

Syntax:

```
$RVS_HOME/bin/rvsGUI -r -u <user> -p <password>
```

The following window is opened and you can login to the server:

Required specifications:

– `User Name`: Login to rvsEVO (to the server) with the user set up in the user management (user ID)

– `Password`: Password of the user

– `Server Host Name`: Name or IP-address of the remote computer and port, to which the computer adheres to. The following syntax is to be considered: `<RMI Server>:<Port>.` Default port is `3755`.

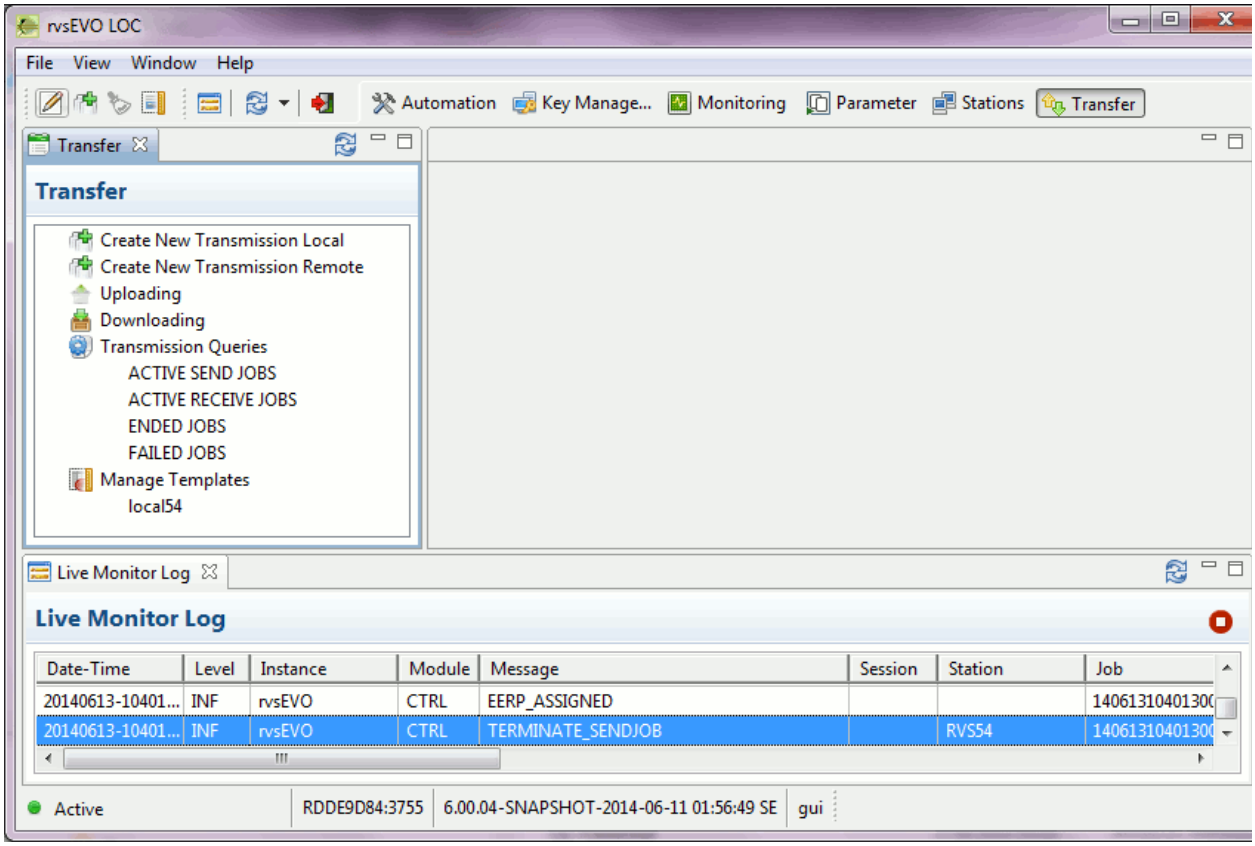
Hint: From the second login on you can select your login data from the drop-down list of the input box `Login History Details.`

4.3.2 Features of Remote GUI

The rvsEVO GUI on the client computer differs marginally from the rvsEVO GUI on the server computer.

The following differences exist:

– In addition, the transfer perspective provides the functionality of creating the sendjob remotely or locally. Thereby, the parameters for the file dispatch remain the same.



– Further, there is an option of uploading (transferring files to the server) or downloading (fetching / transferring files from the server) of files.

4.3.3 Create New Transmission Remote

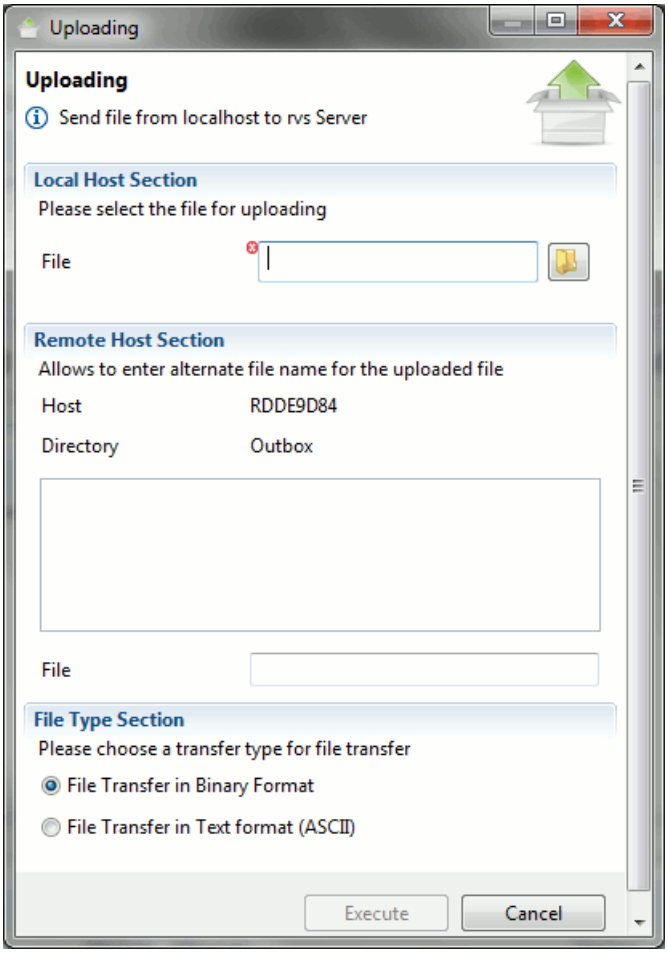
If you create a new transmission remote you can only select files from `outbox` directory. Via `Uploading` entry you can upload the files to be transmitted into `outbox` directory of rvsEVO server.

Uploading

In order to transfer a file to the server (into `outbox` directory), you can select the menu item **uploading** in the transfer tree**.**

The input window is selected in three sections. In `Local Host` Section you can select the file to be sent. The `Remove Host` Section offers the possibility of renaming the file. In `File Type` Section the transmission format can be selected.

Press **Execute** for starting uploading. On rvsEVO Server the file is saved in `$RVS_HOME/files/outbox` directory.



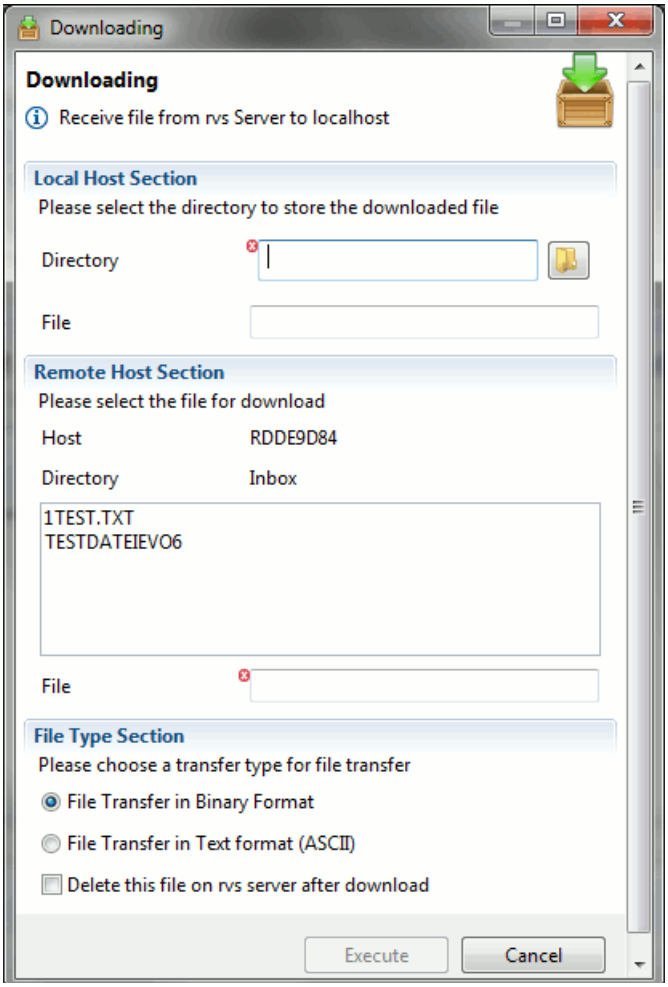
Hint: For the upload the server connects via a port, opened by the client. The port can be set up by the definition `<entry key="client.remotefileloader.port" value="3756" />` in the `$RVS_HOME/conf/rvsEvoClient.prefs` file.

Downloading

In order to receive a file from `inbox` directory of the server, you select the menu item `downloading` in the transfer perspective.

The input window is selected in three sections. In `Local Host` Section you can select the directory where the incoming file is to be saved and you can also rename the file. In `Remove Host` Section you can select the file to be downloaded. In `File Type` Section the transmission format can be selected.

Press **Execute** for starting downloading.



4.3.4 Create New Transmission Local

To create a new transmission local via Remote GUI proceed as described in chapter 6.2 "Create a Send Job". The file to be sent is uploaded into `inbox` directory of the rvsEVO Server automatically.

The **Disposition** and **No Copy** parameters are preset and cannot be edited. **Desposititon** = `D (delete)` means that the file in `outbox` directory is deleted after transmission. **No Copy** means that the rvsEVO Server does not create a second copy of the file.

4.3.5 Command tools

The user ID and password of the user (arranged with the user administration) must be specified for all the command line calls which are started remotely.

Example:

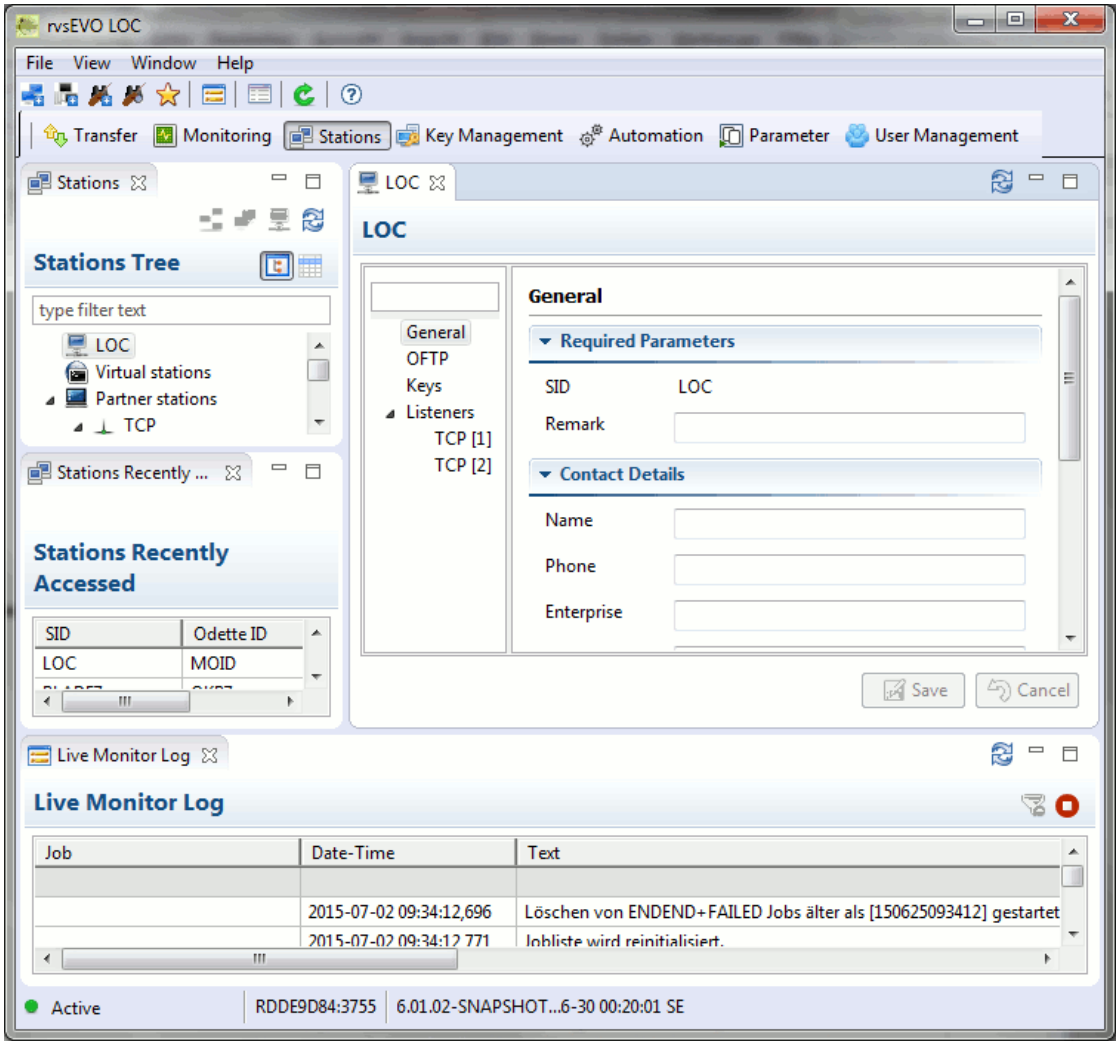
```
activateStation -s <SID> -u <user> -p <password>
```

5 Management of Stations

To be ready for data exchange an entry in the stations configuration is needed for every participated station: your local station, the neighbour stations, routed stations and virtual stations (if needed). The obligatory parameters for the local station have already been polled during installation.

5.1 User Interface of Stations Perspective

The stations management is reachable via menu item `View -> Stations`. The illustration below shows the default view of stations perspective in Standard Edition:



The symbols in the function bar provide the following functionalities:

– Add Neighbour Station

– Add Virtual Station

– Add Proxy Bastion

– Proxy Bastion List
– Stations Recently Accessed
– Show Live Monitor Log
– Active Panel
– Reset Perspective

Thereunder the view `Stations` offers the following functionalities via tool bar:

– Focus on next routed station in current station view (view of neighbour station: Navigation to first routed station; view of routed station: Navigation to the next routed station)
– Focus on neighbour station in current station view
– Focus on local station in current station view
– Reload
– Column configuration
– Minimize
– Maximize

Navigation tree

In the stations view the navigation tree is positioned. All existing stations (local station, virtual stations, Neighbour stations and routed stations) are displayed in the navigation tree. A click on the small arrow to the left of the connection type (TCP/IP, ISDN, XOT, TLS, Proxy TCP/IP or Proxy TLS) - which is reachable via the entry Partner stations - blends or masks out the neighbour stations. Via the arrow to the left of the neighbour stations the routed stations are blended or blanked.

Via the input box above the navigation tree the display can be filtered. If you prompt „rvs“ into the input box, only the stations beginning with „rvs“ are displayed.

Moving the cursor on a station in the navigation tree opens a tooltip displaying the Odette-ID and the remarks.

List view

There are two types of display: navigation tree and list view. The type of display can be selected via the symbols in the tool bar: 

By default, the list view displays the following information: SID, Odette ID, Type, Network, Gateway (only for routed Stations: SID of the neighbour, the routed station is reachable over) and Remark. The parameter can be blended or blanked via a right mouse click on the column header and then selecting the appropriate entry.

With a single-click on the column headings the list entries are sorted.

The list view provides a filter function: Insert the string you want to filter on into the top of the correspondent column.

You can start the following operations with a right mouse click on a station in list view or navigation tree:

Local Station:

– Open in New View
– Add Neighbour Station
– Add Virtual Station
– Add Listener
– Remove Listener

Virtual Station:

– Open in New View
– Duplicate Station
– Delete Entry

Neighbour station and routed Station:

– Open in New View
– Activate Station (only partner station)
– Activate Station without Transfer (only partner station)
– Duplicate Station
– Delete Entry
– Add Routed Station (only partner station)
– Certificate Request to one Station
– Certificate Deliver to one Station

Below the `Stations` view the `Stations Recently Accessed` view is positioned. It contains a list of 10 stations with the latest access.

At the bottom of the GUI the `Live Monitor-Log View` is displayed.

Detailed view of a station

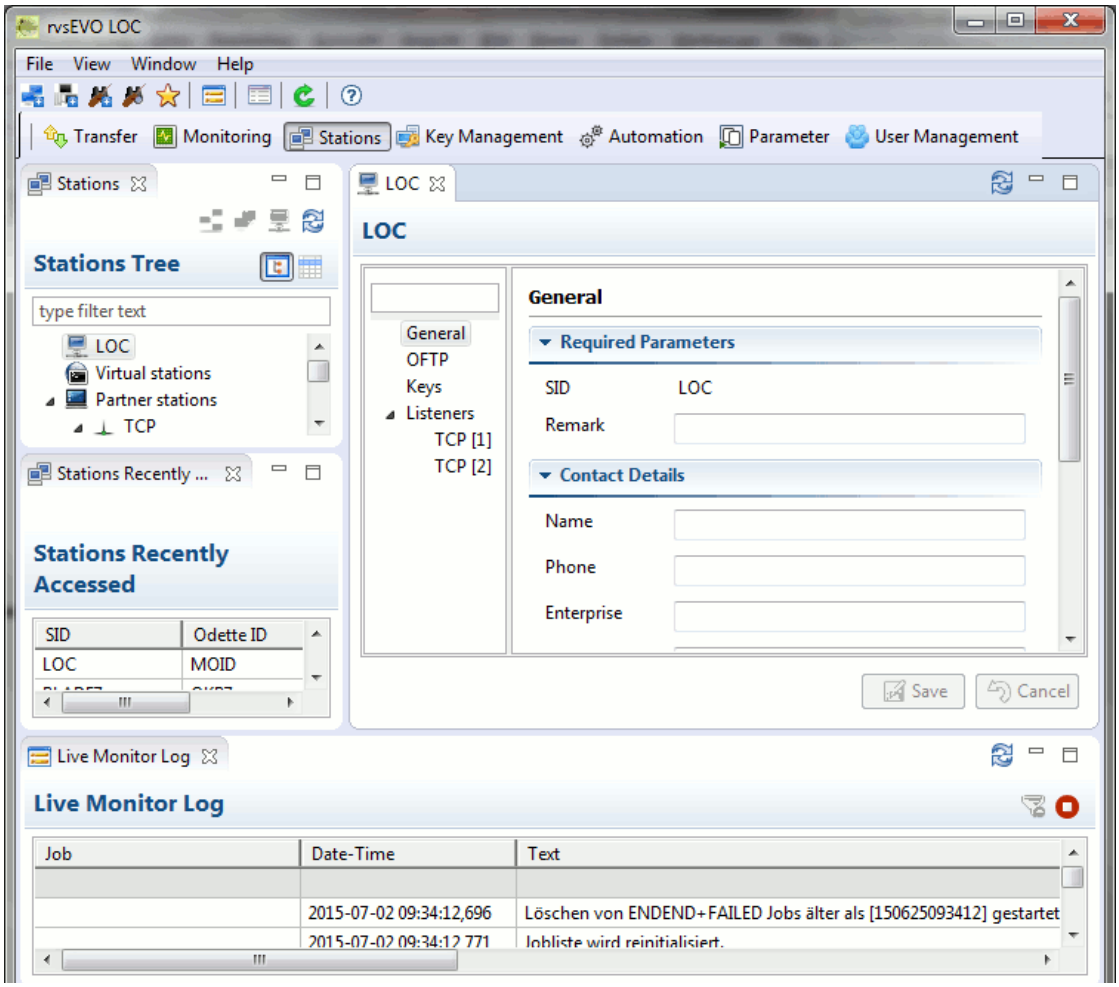
You get detailed information of a station via a single click on this station or via a right mouse click and then selecting `Duplicate View`. The second case several `Detailed Views` are opened at one. The View in the front can be selected via registrations cards. In the detailed view the stations parameters can be edited.

5.2 Local Station

Normally the local station parameters are already set during the installation. In this chapter you find the explanation of all parameters concerning the local station. It is described how to edit the parameters and how to add receivers.

5.2.1 Setting up of a Local Station

Open the detailed view of the local station to set up the parameters.



5.2.2 Parameter Groups of Local Station

There are the following parameter groups: **General**, **OFTP**, **Keys** and **Listeners**. With a single click on the small arrow to the left of listeners the receivers can be blended or blanked.

– **General**: The mandatory parameter is SID. The parameter SID is a locally unique station ID which can consist of up to sixteen characters. It is a strictly local definition; remote stations do not have access to these names. In **Remark** field you can insert a text. The other parameters are contact data. They are optional.

– **OFTP** (ODETTE parameters): For the local station it is only necessary to set up the ODETTE ID. ODETTE ID is a worldwide unique identification of all nodes using the ODETTE file transfer protocol (OFTP). This 25 character name consists of

  – the letter O,

  – an 18 character organization identifier provided by the ODETTE codification group, and

– a 6 character computer sub address that is administrated by each organization.

Note: If you communicate within your own closed network only, the ODETTE ID may be freely chosen as long as it remains unique in your network.

Please see Chapter 22 "PKI Connection" for description of **PKI** and **Certificate Validation Type** parameters.

– **Keys:** In this parameter group the certificates and public keys are listed. You get the following information: Type, Index, Label, Valid From, Valid To, Status. The key you set up at first gets the default label, by default. In the check box `Default Certificate` you can edit this setting.

The block **Key Usage** offers the possibility of functional appropriation.

Key Usage Parameters of Local Station:

Parameter	Description
OFTP2 File Signature	Select the certificate using for file signature by default.
OFTP2 Response Signature	Select the certificate using for EERP/NERP-signature by default.
ComSecure File Signature	Select the certificate using for signature of files which are encrypted via ComSecure

Parameters of Key Usage for TLS/Remote TLS-Listener:

TLS-1: Certificate which is used for TLS encryption by default.

– **Listeners**: Your local station can receive the data of a neighbourstation via several data lines and networks. In this group the already configured listeners (receivers) are displayed.

Add Listener

You can add several listeners of one network or listeners of several networks for your local station. Add a listener via a right mouse click on the entry **Listeners** in detailed view of your local station and then select the network. An other way to add a listener is to right-click the local station in the navigation tree and thereafter select `Add Listener` from the context menu.

Hints:

To add a Proxy TCP/IP or Proxy TLS listener you need to set up a Proxy Bastion before (see chapter Proxy Bastion).

For Windows systems: Unlock the firewall for the ports of the listeners that should be added if the Windows firewall is enabled.

Delete Listener

Right-click the listener to be deleted and then select `Remove Receiver` to delete a listener.

You can also delete a listener by a right mouse click on the local station in the navigation tree, thereafter selection of the option `Remove Receiver` and then selection of the listener.

TCP/IP and Proxy TCP/IP

In the next table you will find the description of the TCP/IP and Proxy TCP/IP parameters:

TCP/IP- and Proxy TCP/IP Parameters (Local)

Parameter	Description
Ordinal Number <ReceiverNumber>	Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of TCP/IP parameters. rvsEVO automatically assigns and manages this number. Default: `1`
Enabled <enabled>	This parameter defines whether or not the TCP/IP listener is enabled or disabled: Default: disabled
IP Address <IPAddress>	IP address or DNS name of the own station. If you own station has only one IP address, you can left this field empty.
Proxy Bastion	Only for Proxy TCP/IP: you can select the Bastion from a list of the Bastion instances created in rvsEVO in which the receiver / listener should be started
Port <Port>	Port on which a TCP/IP listener is to be started; `3305` by default.
Max. Incomming Sessions <Sessions>	Maximal number of simultaneously active receipt processes on the same channel. Maximum: `100`.
Time Out <TimeOut>	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answers. Default: `30` `0` means that the default value is used.

TLS and Proxy TLS

The TLS and Proxy TLS parameters are necessary for the encrypted communication with a partner. TLS (Transport Layer Security) is an encryption protocol for data transmission in an internet. It provides an encryption during the communication way, on the session level.

TLS and Proxy TLS Parameters (Local)

Parameter	Description
Ordinal Number <ReceiverNumber>	Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of TCP/IP parameters. rvsEVO automatically assigns and manages this number. Default: `1`
Enabled <enabled>	This parameter defines whether or not the TLS listener is enabled or disabled: Default: disabled
IP Address <IPAddress>	IP address or DNS name of the own station. If you own station has only one IP address, you can left this field empty.
Proxy Bastion	Only for Proxy TCP/IP: you can select the Bastion from a list of the Bastion instances created in rvsEVO in which the receiver / listener should be started
Port <Port>	Port on which a TCP/IP listener is to be started; `3305` by default.
Max. Incomming Sessions <Sessions>	Maximal number of simultaneously active receipt processes on the same channel. Maximum: `100`.
Time Out <TimeOut>	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answer. Default: `30` `0` means that the default value is used.
Client Authentication <ClientAuthentication>	Sometimes is for the TLS connection necessary, that a client (in this case a partner, from whom you receive data) should be authenticated. With this parameter you can choose, if this is: - `NEEDED` - `NONE` (no authentication) or - `WANTED` **Note**: The authentication should be done with the X.509 certificates.

ISDN

The ISDN parameters for a local station are described in the following table. Please see the chapter 2.1 "System Requirement" for ISDN system requirements. ISDN is only for Windows platform available.

ISDN Parameters (Local)

Parameter	Description
Receiver Number <OrdinalNumber>	Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of ISDN parameters. rvsEVO automatically assigns and manages this number. Default: 1
Enabled <enabled>	This parameter defines whether or not the ISDN listener is enabled or disabled: Default: disabled.
ISDN Address <Address>	An ISDN number of your local station. For the 1TR6 ISDN standard the single-digit (**E**nd**S**election**N**umber) (ESN) is assigned to the local station. Default: " "
Type <DeviceType>	not configurable. Determines the type of connection within your computer - CAPI2. Default: CAPI2A A = no Diehl/Eicon card
Card Number <DeviceCardNumber>	Number of the ISDN card installed in the computer. Default: 1
ISDN Protocol <Protocol>	Specifies the ISDN standard used. - 1TR6: German national standard - E-DSS1: EURO-ISDN Default: E-DSS1
ISDN Facilities <Facilities>	Special information or facilities of the ISDN transmission.
ISDN User Data <Userdata>	User data for the ISDN transmission
ISDN Terminal Identifier <Tei>	Only required for X.31: Terminal End Identification. Default: 0 – no TEI assigned
Time Out <Timeout>	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answers. Default: 60
RCV Time Out <RestartTimeOut>	Only valid for the local station: Cancellation time in miliseconds, after which the receiver interrupts the waiting for incoming calls and reconnects to the card driver Default: 0

ISDN Parameters (Local)

Parameter	Description
Max. Incomming Sessions <Sessions>	Maximal number of simultaneous active receipt processes on the same channel. Default: 1 Maximum: 100
X.25 Address <X25Address>	15-digit X.25 DTE address of the local station. This number is optional. It is however advisable for an ISDN connection to enter the ISDN number here, as some partners expect an X.25 address.
X.25 PacketSize <PacketSize>	Size of data packets during data transmission Default: 128
X.25 WindowSize <WindowSize>	Window size in X.25/ISDN communication is the number of packets that can be outstanding without acknowledgment. The window size will be negotiated during connection setup, but we recommend you to use the correct window size (depending on the partner network). Possible values: 1 - 7 (default: 7) We recommend 7 for ISDN and 2 for X.25 native
X.25 Facilities <X25Facilities>	Special information or facilities of the ISDN transmission.
X.25 User Data <X25UserData>	User data for the ISDN transmission
X.25 Dbit <DBit>	D-bit: Delivery Confirmation;an X.25 data packet flag used to request end-to-end acknowledgment for the packet.
X.25 Closed User Group <ClosedUserGroup>	SDN and X.25 allow for the formation of a closed user group. All members of such a group can communicate with each other via the public telecommunication network. Connection requests to group members received from participants not being a member of the closed user group will be rejected by the switching exchange. The same applies to connection requests from group members to participants not being a member of the closed user group. This service attribute is called Closed User Group (CUG).

XOT

XOT (X.25 over TCP/IP) routers are able to route X.25 packets between a TCP/IP network on one side and an X.25 or ISDN network on the other side.

System requirements:

Please read the chapter “XOT systems requirement” on page 19.

The XOT parameters for a local station are described in the following table.

XOT Parameters (Local)

Parameter	Description
Ordinal Number <OrdinalNumber>	Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of ISDN parameters. rvsEVO automatically assigns and manages this number. Default: 1
Enabled <enabled>	This parameter defines whether or not the XOT listener is enabled or disabled: Default: disabled.
Local IP address	Local station only (optional): own IP address. IP addresses have the form "255.255.255.255". If you have not specified a value for the local station, you permit automatic definition of the IP address. If your own station has only one IP address, this field should be left empty.
Local Port <LocalPort>	Port of the local station Default: 1998 for XOT communication.
X.25 Address <X25Address>	15-digit X.25 DTE address of the local station. This number is optional. It is however advisable for an ISDN connection to enter the ISDN number here, as some partners expect an X.25 address.
Max. Incomming Sessions <Sessions>	Maximal number of simultaneous active receipt processes on the same channel. Default: 1 Maximum: 100
Time Out <Timeout>	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answers. Default: 30 0 menas that default value is used.
Router IP Address <RouterIPAddress>	IP address of an XOT router. This parameter is optional for the local station and mandatory for the partner station
Router Port <Port>	Port of the XOT router Default: 1998 for XOT communication
X.25 PacketSize <PacketSize>	Size of data packets during data transmission Default: 128

XOT Parameters (Local)

Parameter	Description
X.25 WindowSize <WindowSize>	Window size in X.25/ISDN communication is the number of packets that can be outstanding without acknowledgment. The window size will be negotiated during connection setup, but we recommend you to use the correct window size (depending on the partner network). Possible values: 1 - 7 (default: 7) We recommend 7 for ISDN and 2 for X.25 native.
X.25 Facilities <X25Facilities>	Special information or facilities of the ISDN transmission.
X.25 User Data <X25UserData>	User data for the ISDN transmission
X.25 Dbit <DBit>	D-bit: Delivery Confirmation;an X.25 data packet flag used to request end-to-end acknowledgment for the packet.
X.25 Modulo <X25Modulo>	In X.25 data transmission exist two modulo operating standards: Modulo 8 and Modulo 128. Modulo 128 means, that you must select bigger window size via parameter **X.25 Facilities**. The standard window size for modulo 8 is 2. Default: 8

5.3 Neighbour Stations

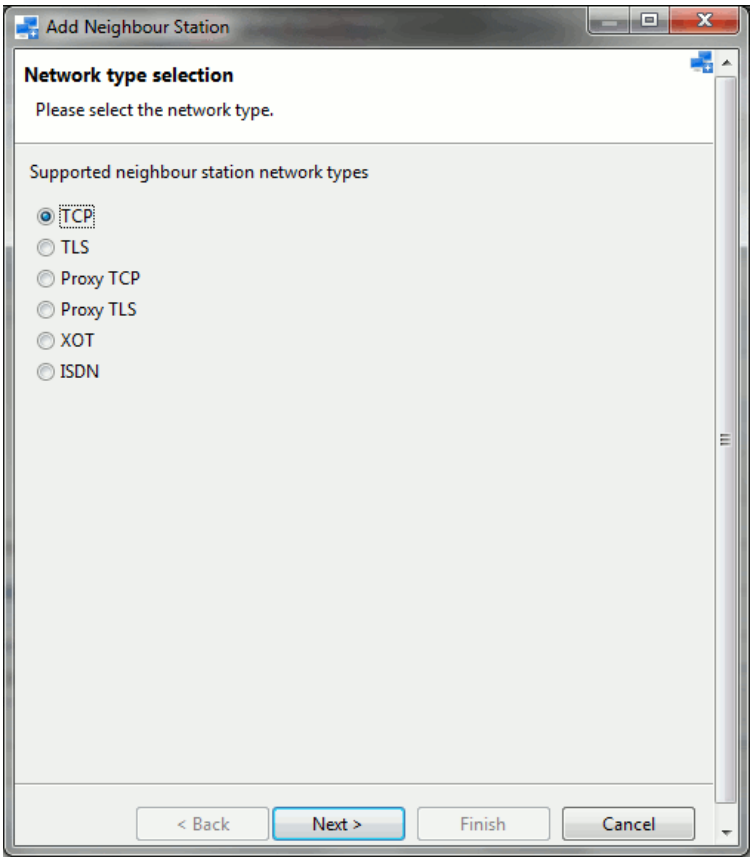
This chapter shows how to add or delete a neighbour station and how to set up the parameters of a neighbour station.

5.3.1 Add a Neighbour Station

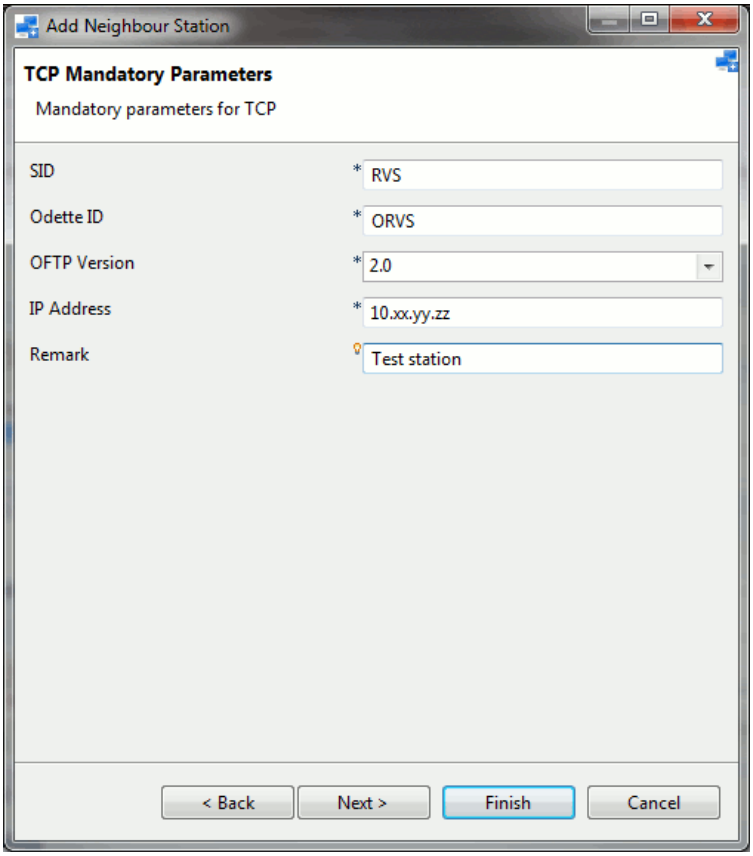
Follow the description below to add a neighbour station:

- Right-click the local station and select `Add Neighbour Station` from the drop-down menu or click the **Add Neighbour Station** icon in the function bar.
- Select a network for the communication. Possible are TCP/IP, TLS, Proxy TCP/IP, Proxy TLS, XOT and ISDN. A TLS connection is

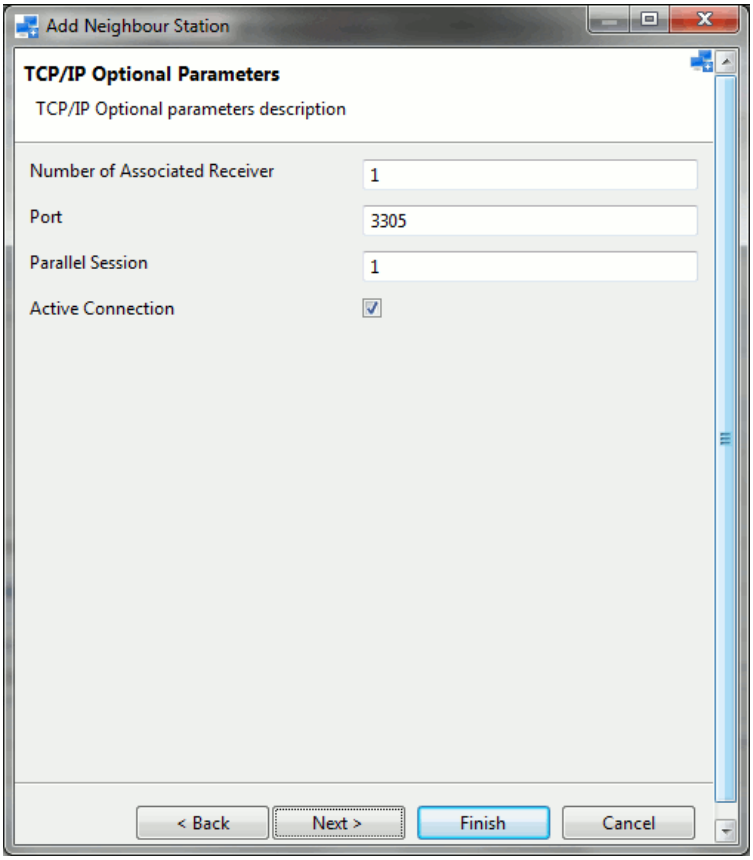
needed for line encryption. Click **Next** to continue.



- In the next window you have to insert the station ID, Odette ID, OFTP Version, IP Address (or hostname) of your neighbour station. In Remark field you can insert a text. Click **Finish** to add the neighbour station or **Next** to insert the optional parameters.



- If needed, define the optional parameters and click **Finish** to add the neighbour station or **Next** to insert the optional network parameters.



Hint: If you selected an ISDN network you have the possibililty to set up the X.25 parameters in the next window.

The following tables insert the description of the network parameters:

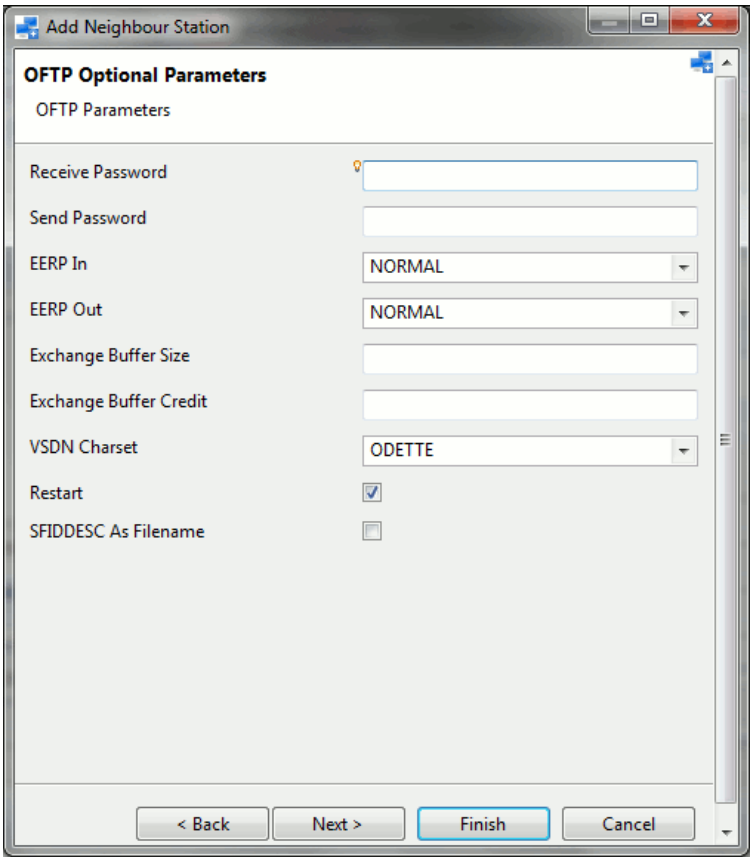
– “TCP/IP, TSL, Proxy TCP/IP and Proxy TLS Parameters (Partner)” on page 66

– “ISDN Parameters (Partner)” on page 66

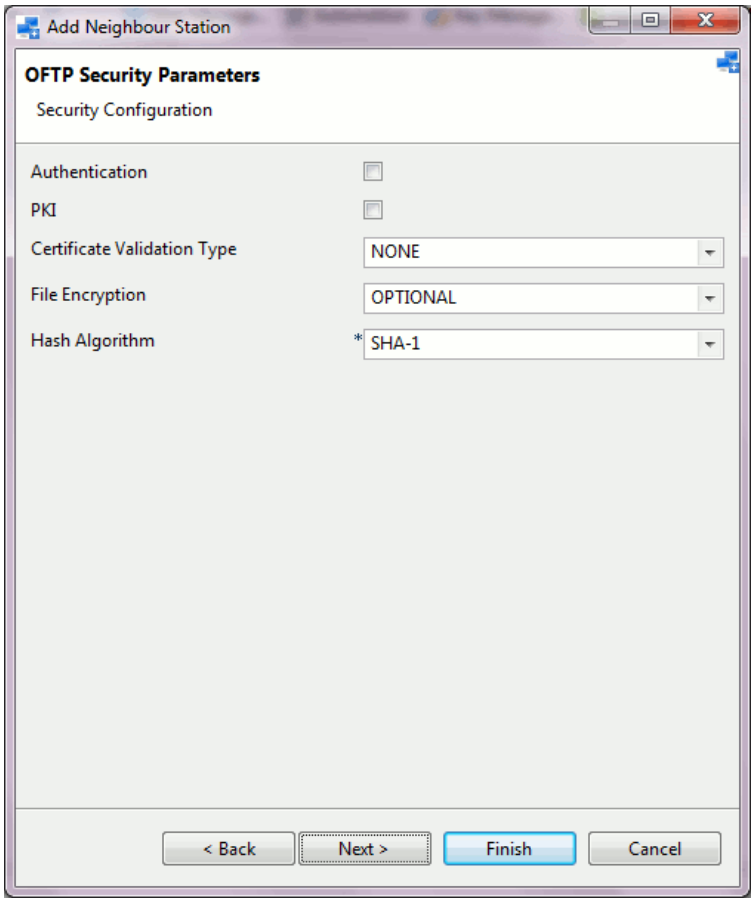
– “XOT Parameters (Partner)” on page 68

- The dialog below offers the possibility to set up the OFTP Optional Parameters. Click **Finish** to add the neighbour station or **Next** to contiue. (You can find the description of the OFTP parameters in table

“Optional OFTP Parameters (Partner)” on page 62)

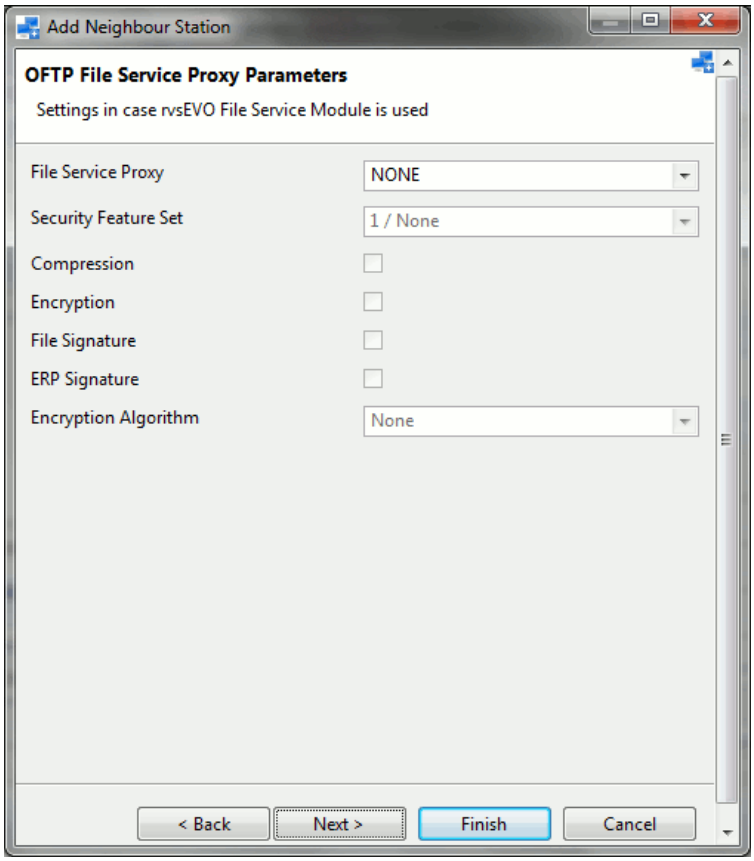


- The next dialog offers the possibility to set up the OFTP security parameters. Click **Finish** to add the neighbour station or **Next** to continue.

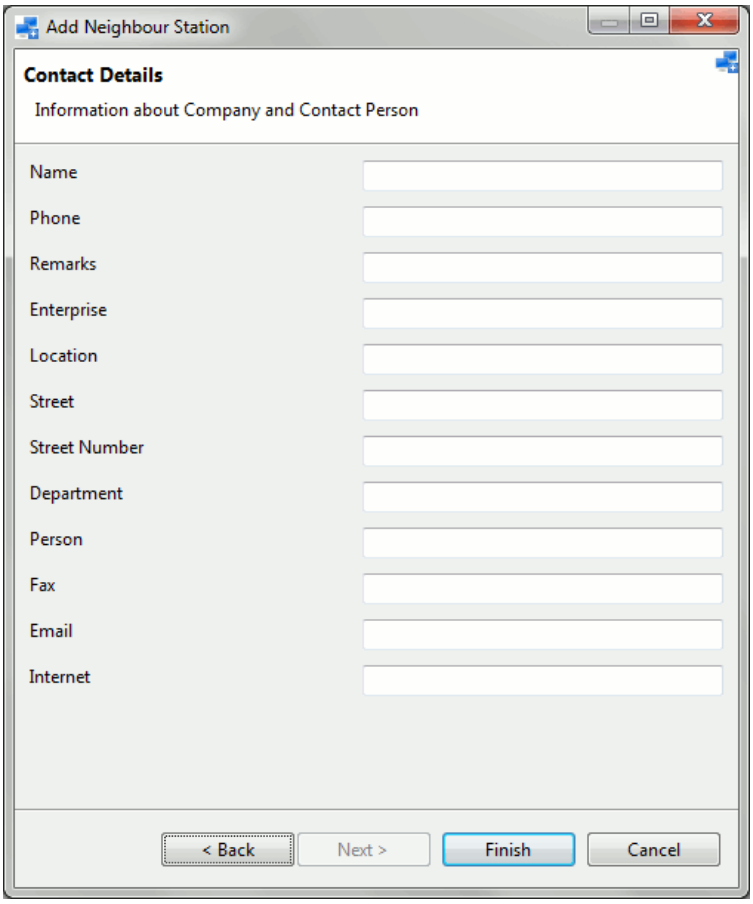


- The next dialog offers the possibility to set up the parameters for OFTP File Service Proxy. Click **Finish** to add the neighbour station or **Next** continue.

.



- The last parameters are contact data. Click **Finish** to add the neighbour station.



Hint:

You can add a neighour station with identical settings to an existing station. Right-click the neighbour station that is to be duplicated and select `Duplicate Station` entry from the combo box. The settings of the duplicated stations are used as default values of the new station. SID and Odette ID parameters must be changed. The remaining parameters can also be edited if neccesary.

5.3.2 Setting up of a Neighbour Station

Open the detailed view for setting up of a neighbour station. Edit the parameters and then click **Save**.

5.3.3 Parameter Groups of the Partner Station:

There are the following parameter groups: **General**, **OFTP**, **Keys** und **Listeners**. With a single click on the small arrow to the left of listeners the receivers can be blended or masked out.

– **General**: The mandatory parameters are SID and Network Type. The parameter SID is locally unique station ID which can consist of up to

sixteen characters. ’A-Z’, ’0-9’, ’-’, ’_’ and ’.’ are possible characters. ’.’ may not rank first. SID is a strictly local definition; remote stations do not have access to these difinitions; they only know the ODETTE IDs. The network parameter was assigned during adding a neigbour station. In **Remark** field you can insert a text. The other parameters are optional contact data.

– **Keys:** This group displays the certificates what are available for this station. At the top of the windows all certificates are listed. You get the following information: Type (= Private Key, =Partner Certificate, = Intermediate Certificate, =CA Certificate), Index, Label, Valid From, Valid To, Status.
Via the input box `Default Key` the Key with default label can be defined.

Import certificates

Via the buttons **Import Certificate** and **Import Comsecure Certificate** the certificates of your partner can be imported. The certificate which was imported at first gets the default label, by default. In the check box `Default Certificate` you can edit this setting.

The block **Key Usage** offers the possibility of functional appropriation.

Key Usage Parameters (Partner)

Parameter	Description
OFTP2 File Signature	Select the certificate using for file signature by default.
OFTP2 Response Signature	Select the certificate using for EERP/NERP-signature by default.
ComSecure File Signature	Select the certificate using for signature of files which are encrypted via ComSecure

The next parameters only have to be set if you wish to use another certificate than that one with default label.

Certificate Usage Parameters (Partner):

Parameter	Description
OFTP2 Authentication	Select the certificate of your partner which you want to use for authentication.
OFTP2 File encryption	Select the certificate of your partner which you want to use for file encryption.
ComSecureFile encryption	Select the certificate of your partner which you want to use for ComSecure encryption.

– **OFTP**: This group contains the parameters which refer to ODETTE protocol. In the following table you will find detailed explanation of ODETTE group parameters.

Required OFTP Parameters (Partner):

Parameter	Description
Odette ID <OdetteID>	ODETTE ID is a worldwide unique identification of all nodes using the ODETTE file transfer protocol (OFTP). This 25 character name consists of - the letter 0, - an 18 character organization identifier provided by the ODETTE codification group, and - a 6 character computer sub address that is administrated by each organization.
OFTP Version <Level>	which OFTP version should be used. Possible values: 1.3, 1.4 and 2.0

Optional OFTP Parameters (Partner)

Parameter	Description
Receive Password <PasswordReceive>	The password rvsEVO expects from the neighbour station.
Send Password <PasswordSend>	The password rvsEVO sends to the neighbour. ODETTE password exchange between two neighbour stations and verification always occurs while a sessio n is being established. **Hint:** With `activateStation` tool (chapter 12.3.1) this password can be overwritten temporarily for the unique connection.
EERP in <EERPIn>	Procedure for receiving receipts (EERPs/ NERPs). Possible values: `NORMAL`: The transmission is successfully terminated after receiving of EERP. `NEVER`: no EERP is needed. The transmission can be terminated without receiving a receipt. **Hint**: some operator use EERP destination and EERP originator in a different way. This causes problems with the identification of EERPs. Set the parameter **rvs_evo.job.additional_erp_reverse_search** =true in `$RVS_HOME\conf\rvs-system.properties` file to resolve this problem. Default: `NORMAL`

Optional OFTP Parameters (Partner)

Parameter	Description
EERP Out <EERPOut>	Procedure for sending receipts (EERPs/NERPs). NORMAL: Generation of a receipt after successful file reception and immediate active transmission. NEVER:no Generation of a receipt. The job is changed to ENDED status. HOLD: Generation of a receipt after successful file reception. The receipt, however, is only sent after having been released with the handleEERP program. SYNC:The OFTP session between sender station and receiver station will be kept alive waiting for an EERP, that should come from the receiver station. ROUTING_SYNC: **Hint**: This parameter is important only for a routing station. The OFTP session between the router station and the sender station will be kept alive waiting for an EERP, that should come from the receiver station. When the EERP is arrived, it will be sent in the same session to the sender. Default: NORMAL (see also parameter **End to End Response in**)
Exchange Buffer Credit <Credit>	Maximal number of sent blocks (Exchange Buffer) without an acknowledgment. Possible values: 1 - 999 (default: 999). No entry means that the default value is used.
Exchange Buffer Size <BufferSize>	Maximal size of ODETTE Exchange Buffer in Bytes. Possible values: 1 - 99999 (default: 10.000). No entry means that the default value is used. Attention: With ISDN connection the value should not be over 4.000 bytes since otherwise the transmission of large files can cause problems.
VDSN charset <VdsnCharset>	For receive jobs only: allowed character set for VDSN. Possible values: - ODETTE: ODDETTE character set is allowed only (default). - ALL: all ASCII characters are allowed
Restart <restart>	This parameter defines whether a restart is allowed after an error message. Possible values: true (default) or false:
SFIDDESC as Filename <UseDescAsFilename>	SFID (Start File ID) is used as filename (instead of VDSN) for the transmisson.

OFTP Security Parameters (Partner):

Parameter	Description
Authentication <Authentication>	Authentification on ODETTE level at the beginning of the transmission. Possible values: - `Yes` (X.509 certificate has to be exchanged) - `No`
PKI	Please read chapter 22.2 "Configuration of PKI Binding" for detailed information.
Certificate Validation Type <CertificateValidationType>	Please read chapter 22.2 "Configuration of PKI Binding"

File Encryption <SecuritySet>	Defines whether or not encryption is to be used during file transmission. Values:
	`File Encryption=NO`
	Encryption is impossible. The job aborts with an error message if a send job requires encryption.
	`File Encryption=OPTIONAL`
	Encryption possible as an option and can be specified in the send job.
	`File Encryption=FORCED`
	Encryption is compulsory. If a send job is scheduled without encryption a warning is issued. If a partner station sends an unencrypted file, reception of this file is refused.
	Default: `SECURITY=OPTIONAL`
	Hint: By default, this parameter refers to transmission via OFTP2, only. If this setting should be valid also for ComSecure the `rvs_evo.controller.check_security_by_sfid=false` dedinition of the `$RVS_HOME/conf/rvs-system.properties` file has to be set.

Hash Algorithm	The following values can be selected: - `SHA-1` (default) - `SHA-256` - `SHA-512`

OTFTP File Service Proxy Parameters (Partner):

Parameter	Description
File Service Proxy <Proxy>	Only for File Service Modul; possible values: - NONE (default) - INTERNAL - EXTERNAL Please see chapter 18 "File Service Module" for detailed explanation.
Security Feature Set <SecurityFeatureSet>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.
Compression <Compression>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.
Encryption <Encryption>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.
File Signature <Sign>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.
ERP Signature <SignERP>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.
Encryption Algorithm <EncryptionAlgorithm>	Send parameter on station level. Only available if parameter **File Service Proxy = EXTERNAL**. Please see table “Security parameters:” on page 84 for detailed explanation.

– **Line Type**: The only parameters in this group are **Parallel Session** and **Active Connection**. In **Parallel Session** you indicate the max. Number of Sessions to this station at the same time.
In **Active Connection** parameter you define if your local station establishs the connection automatically or not. When activating the checkbox the connection to the station is set up automatically if a sendjob is started. On the contrary, the partner station is to establish the connection and fetch the available files when required. The partner bears the costs for the connection.

– The selected network: **TCP/IP**, **TLS**, **ISDN**, **XOT**, **Proxy TCP/IP** or **Proxy TLS**. In the following tables you can find the explanation of the network parameters

TCP/IP, TSL, Proxy TCP/IP and Proxy TLS Parameters (Partner)

Parameter	Description
IP Address	IP address or DNS name of the neighbour station.
Number of Associated Receiver	Not active! Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of ISDN parameters. rvsEVO automatically assigns and manages this number. Default: 1
Port	Port of neighbour station; default: 6619.
Proxy Bastion	for rvs® OFTP Proxy only; select the Bastion Instance which should be used for communication

ISDN

In the following table you can find the explanation of the ISDN parameters.

ISDN Parameters (Partner)

Parameter	Description
ISDN Address	An ISDN number of the partner station.
ReceiverNumber	Not active! Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of ISDN parameters. rvsEVO automatically assigns and manages this number. Default: 1
Type	Determines the type of connection within your computer - CAPI2. Default: CAPI2A A = no Diehl/Eicon card
Card Number	Number of the ISDN card installed in the computer. Default: 1
ISDN Protocol	Specifies the ISDN standard used. - 1TR6: German national standard - E-DSS1: EURO-ISDN Default: E-DSS1
ISDN Facilities	Special information or facilities of the ISDN transmission.

ISDN Parameters (Partner)

Parameter	Description
ISDN Userdata	User data for the ISDN transmission
ISDN Terminal Identifier	Only required for X.31: **T**erminal **E**nd **I**dentification. Default: `0` – no TEI assigned
Time Out	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answers. Default: 60
Dial Retry Count	Count of dial retries, if the partner does not answer
Dial Retry Wait Time	Wait time between two dial retries, if the partner does not answer
Number of Associated eceiver <ReceiverNumber>	Number for differentiating the various receiving channels through which the local station can be reached. Each number has a tab with a set of ISDN parameters. rvsEVO automatically assigns and manages this number. Default: `1`
X.25 Address <Address>	15-digit X.25 DTE address of the partner station.
X.25 Closed User Group	ISDN and X.25 allow for the formation of a closed user group. All members of such a group can communicate with each other via the public telecommunication network. Connection requests to group members received from participants not being a member of the closed user group will be rejected by the switching exchange. The same applies to connection requests from group members to participants not being a member of the closed user group. This service attribute is called Closed User Group (CUG).
X.25 DBit	D-bit: Delivery Confirmation; an X.25 data packet flag used to request end-to-end acknowledgment for the packet.
X.25 Facilities	Special information or facilities for an X.25 transmission; see the information of the X.25 service in use
X.25 PacketSize <PacketSize>	Size of date packets during data transmission
X.25 Userdata	User data for an X.25 transmission; see the information of the X.25 service in use.

ISDN Parameters (Partner)

Parameter	Description
X.25 Window Size <WindowSize>	window size 7 (recommended for ISDN) window size 2 (recommended for X.25 native) **Note**: Window size in X.25/ISDN communication is the number of packets that can be outstanding without acknowledgment. The window size will be negotiated during connection setup, but we recommend you to use the correct window size (depending on the partner network).

XOT The XOT parameters are described in the table below.

XOT Parameters (Partner)

Parameter	Description
Number of Associated Receiver <ReceiverNumber>	Not active! Remote station only: Number for differentiating the various receivers through which the local station can be reached. Note: Each receiver should be defined in one XOT tab of the local station. rvsEVO automatically assigns and manages this number. Default: `1`
Router IP address	IP address of an XOT router. This parameter is optional for the local station and mandatory for the partner station
Router Port	Port of the XOT router Default: `1998`
Time Out <TimeOut>	Time Out in miliseconds after which the communication program the connection closes, if the partner station does not answers. Default: `30` `0` means that default value is used.
X.25 Address	15-digit X.25 DTE address of the local station. This number is optional. It is however advisable for an ISDN connection to enter the ISDN number here, as some partners expect an X.25 address.
X.25 DBit	D-bit: Delivery Confirmation; applies only to Data packets. It is used to ensure that acknowledgement of Data packet is end-to-end. When not set, a local node can acknowledge the receipt of a Data Packet before forwarding it on the next node.
X.25 Facilities	Special information or facilities for an X.25 transmission; see the information of the X.25 service in use

XOT Parameters (Partner)

Parameter	Description
X.25 Modulo	In X.25 data transmission exist two modulo operationg standards: Modulo 8 and Modulo 128. Modulo 128 means, that you must select bigger window size via parameter FACILITIES. The standard window size for modulo 8 is 2.
X.25 PacketSize <PacketSize>	Size of date packets during data transmission
X.25 Userdata	User data for an X.25 transmission; see the information of the X.25 service in use
X.25 Window Size <WindowSize>	window size 7 (recommended for ISDN) window size 2 (recommended for X.25 native) **Note**: Window size in X.25/ISDN communication is the number of packets that can be outstanding without acknowledgment. The window size will be negotiated during connection setup, but we recommend you to use the correct window size (depending on the partner network).

5.3.4 Delete a Neighbour Station

Right-click the neighbour station to be deleted and select `Delete Entry` to delete a station.

5.3.5 Activate a Neighbour Station

Right-click the station and select `Activate Station` to activate the station and receive files from this station. You need this functionalty to get files from your neighbour station if parameter **Active Connection** (group **Line Type**) is disabled on neighbour station side.

For testing the connection to a neighbour station without file transmission use the option `Activate Station without Transfer` in the drop-down menu.

Hint: You can also use the `activateStation` script to activate a station.

5.3.6 Automatic Certificate Exchange

Deliver certificate request

Right-click the neighbour station and select `Certificate Reqest to one Station`. In the opening window select the station ID of the sender and receiver and the certificate which should be delivered to your partner station. Please see also chapter 8.4.12 "Automatic Certificate Exchange via rvsEVO-GUI".

Deliver certificate

Right-click the neighbour station and select `Certificate Deliver to one Station` to send a certificate to your neighbour station. In the opening window select the station ID of the sender and the receiver and

the certificate which should be dilivered to your partner station. Please see also chapter 8.4.12 "Automatic Certificate Exchange via rvsEVO-GUI".

5.4 Routed Stations

A routed station is reachable via a neighbour station. For this reason you must already have set up a direct neighbouring node about which the routed station is to be added.

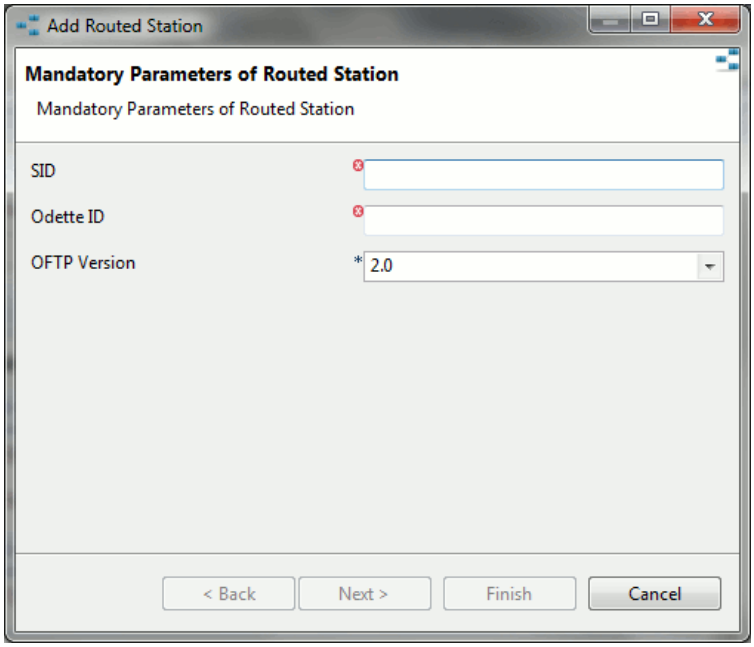
5.4.1 Add a Routed Station

Add a routed station by a right-click on the neighbour station and thereafter selecting the item `Add routed station` from the context menu.

The dialog box below opens and you can set up the mandatory parameters:

– **SID** (Station ID of routed Station) and

– **Odette ID** (please see OFTP parameter group in chapter 5.3.3 "Parameter Groups of the Partner Station:".)



There other parameter groups of routed stations are **OFTP Optional Parameters, OFTP File Service Proxy Parameters** and **Contact Details.**

Please see Chapter 22 "PKI Connection" for getting information of **PKI** and **Certificate Validation Type** Parameters.

The description of the other parameters you can find in chapter 5.3.3 "Parameter Groups of the Partner Station:".

Hint:

You can add a routed station with identical settings to an existing station. Right-click the routing station that is to be duplicated and select `Duplicate Station` entry from the combo box. The settings of the duplicated stations are used as default values of the new station. SID and Odette ID parameters must be changed. The remaining parameters can also be edited if neccesary.

5.4.2 Setting up a Routed Station

Please see chapter 5.3.2 "Setting up of a Neighbour Station".The procedure is the same with all stations.

5.4.3 Further Functionalities

A right-click opens a drop-down menu which offers the following additional options: `Open In New View, Delete Entry, Certificate Request to one Station and Certificate Deliver to one Station`. The description of this functionalities you can find in chapter 5.3 "Neighbour Stations".

5.5 Virtual Stations

Virtual stations are used to represent stations outside the OFTP network for the OFTP network. They allow files to be sent to destination stations outside the OFTP network.

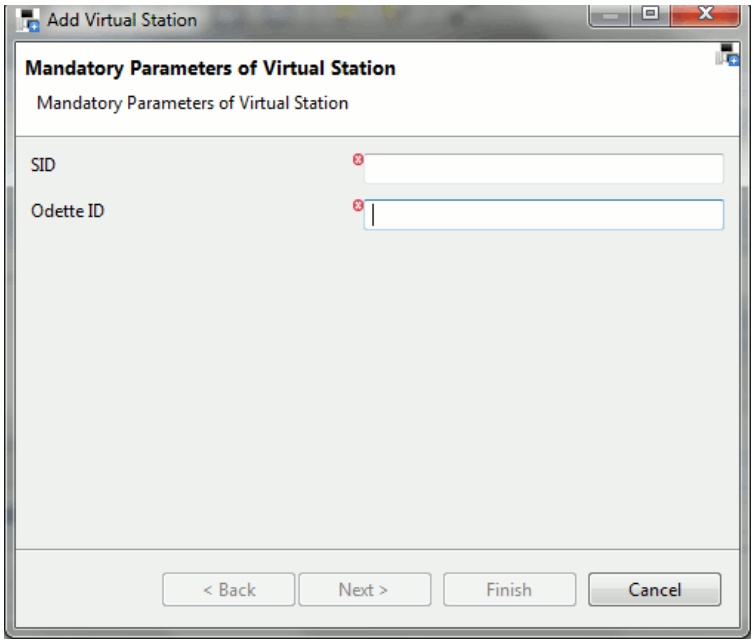
ODETTE-IDs uniquely define the stations in an OFTP network. Each virtual station must also be assigned an own ODETTE ID. A virtual station can also send and receive files.

5.5.1 Add a Virtual Station

Add a virtal station by a right-click on the local station and thereafter selecting the item `Add Virtal Station` from the context menu. The dialog box below opens and you can set up the **Mandatory Parameters**:

– **SID** (Station ID of virtual `Station) and`

– `Odette ID` (please see OFTP parameter group in chapter 5.2.2 "Parameter Groups of Local Station".)



The other parameter groups of virtual stations are **OFTP Optional Parameters** and **Contact Details.**

Please see Chapter 22 "PKI Connection" for getting information of **PKI** and **Certificate Validation Type** Parameters.

The description of the other parameters you can find in chapter 5.3.3 "Parameter Groups of the Partner Station:".

Hint:

You can add a virtual station with identical settings to an existing station. Right-click the virtual station that is to be duplicated and select `Duplicate Station` entry from the combo box. The settings of the duplicated stations are used as default values of the new station. SID and Odette ID parameters must be changed. The remaining parameters can also be edited if neccesary.

5.5.2 Setting up a Virtual Station

Please see chapter 5.3.2 "Setting up of a Neighbour Station".The procedure is the same with all stations.

5.5.3 Further Functionalities

A right-click opens a drop-down menu which offers the following additional options: `Open In New View, Duplicate Station, Delete Entry`. The description of this functionalities you can find in chapter 5.3 "Neighbour Stations".

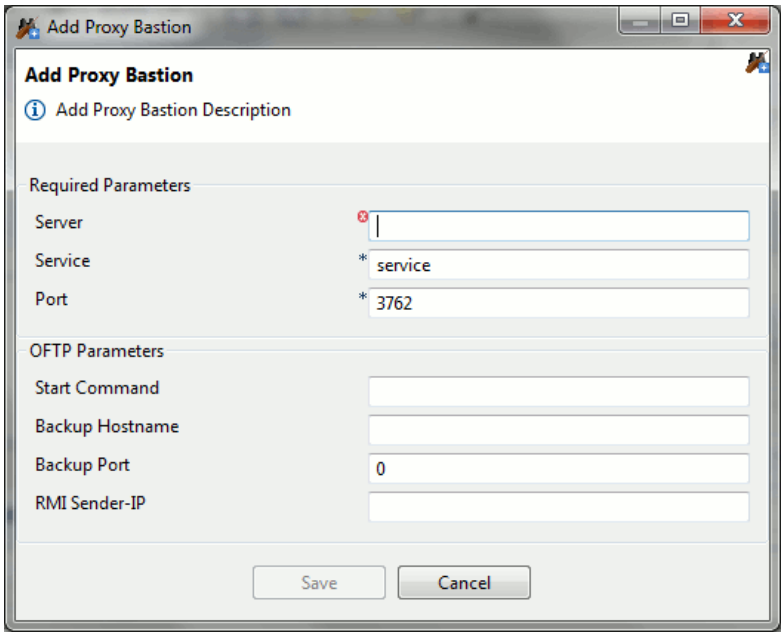
Hint: At the partner station, the treatment of a virtual station is the same as the treatment of a routed station.

5.6 Proxy Bastion

You need to add a Proxy Bastion instance as pre-condition for using the functionality rvs® OFTP Proxy.

5.6.1 Add Proxy Bastion

Click on `Add Proxy Bastion` icon in the function bar. The dialog box below opens and you can edit the parameters. Thereafter the rvsEVO server should be restarted.



The parameters to be entered have already been saved during the installation of Bastion Instance.

In the following table you find the description of Proxy Bastion parameters:

Required Proxy Bastion Parameters

Parameter	Description
Server	Name or IP-address of the server, on which the Bastion Instance has been installed. When using NAT (Network Address Translation) procedure the public IP address.
Service	Name of the Bastion Service. Default: service
Port	Port of the Bastion Instance for the RMI-communication, on which a listener is started.

Optional Proxy Bastion Parameters

Parameter	Description
Start Command	Directory and Start Command of rvs® OFTP Proxy
Backup Hostname	Name or IP address of a machine with a back-up installation of Proxy Bastion.
Backup Port	Port for RMI communication of the Backup installation on which a listener is started.
RMI Sender-IP	Hostname or IP address of the sender used by rvsEVO server for connection with the Proxy Bastion.

5.6.2 Display Proxy Bastions

Click the `Proxy Bastion List` icon in the function bar to display all configured Proxy Bastion instances. The list is output in a new view.

Via double-click on an entry of the Proxy Bastion list you get the detailed view of the Proxy Bastion.

5.6.3 Setting up Proxy Bastion

Open the detailed view for setting up the parameters and confirm your changings with a click on the button **Save**.

Restart rvsEVO server to enable the changes.

Important: It is to be considered, that these parameters are customised also in the start script `$RVS_HOME/bin/bastion.cmd` (`$RVS_HOME/bin/bastion.sh`).

5.6.4 Delete Proxy Bastion

Right-click the Proxy Bastion in the list view and select the item `Delete Proxy Bastion` from the context menu to remove a Proxy Bastion. Another possibility to remove a Proxy Bastion is to assign the Proxy Bastion in the list view and then click the icon `Delete Proxy Bastion` in the function bar.

5.7 Station Configuration via rvsStationlist.xml

On the other hand the station configuration can be done via the XML station configuration file `$RVS_HOME/conf/rvsStationlist.xml`. The `StationLoc` element in the configuration file is equivalent to the Local Station in the GUI, `StationNeighbour` is Neighbour Station, `StationRouted` is the Routed Station and `StationVirtual` is the virtual station in the GUI.

Abstract of `rvsStationlist.xml` file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RVS_STATION_CONFIG>
-<StationLoc>
 . . .
-</StationLoc>
-<StationNeighbour>
 . . .
-<StationNeighbour>
-<StationRouted>
. . .
-</StationRouted>
-<StationVirtual>
. . .
-</StationVirtual>
</RVS_STATION_CONFIG>
```

Lines starting with (<!--) and ending with (-->) are interpreted as comments.

Edit the `rvsStationlist.xml` file if you have to change mandatory parameters (ODETTE ID, TCPIP_Basic) or if you wish to assign values to other optional parameters such as Contact.

Use a text editor (e.g. Edit, TextPad) to do so. Please make sure to save your XML files as valid XML documents after editing as otherwise rvsEVO will not be able to read them and may fail to start correctly.

The changes in the GUI will be visible immediately (after saving) in the XML station configuration file; the changes in the XML station configuration file will be visible in the GUI only after a new start of the rvsEVO (command startGUI).

Please see chapter 12.2 "updateStationList" for getting information about setting up the stations at run time.

Note: For correct TCP/IP communication you must ensure that the IP ports for RMI (1099) and for Odette (e.g. 3305) are free.

For the description of the parameters in the `rvsStationlist.xml` file please refer to the description of the GUI parameters.

The parameter label is not configurable via the GUI:

Parameter of rvsStationList.xml

Parameter	Description
-Label	Free text; use this parameter for labeling the station

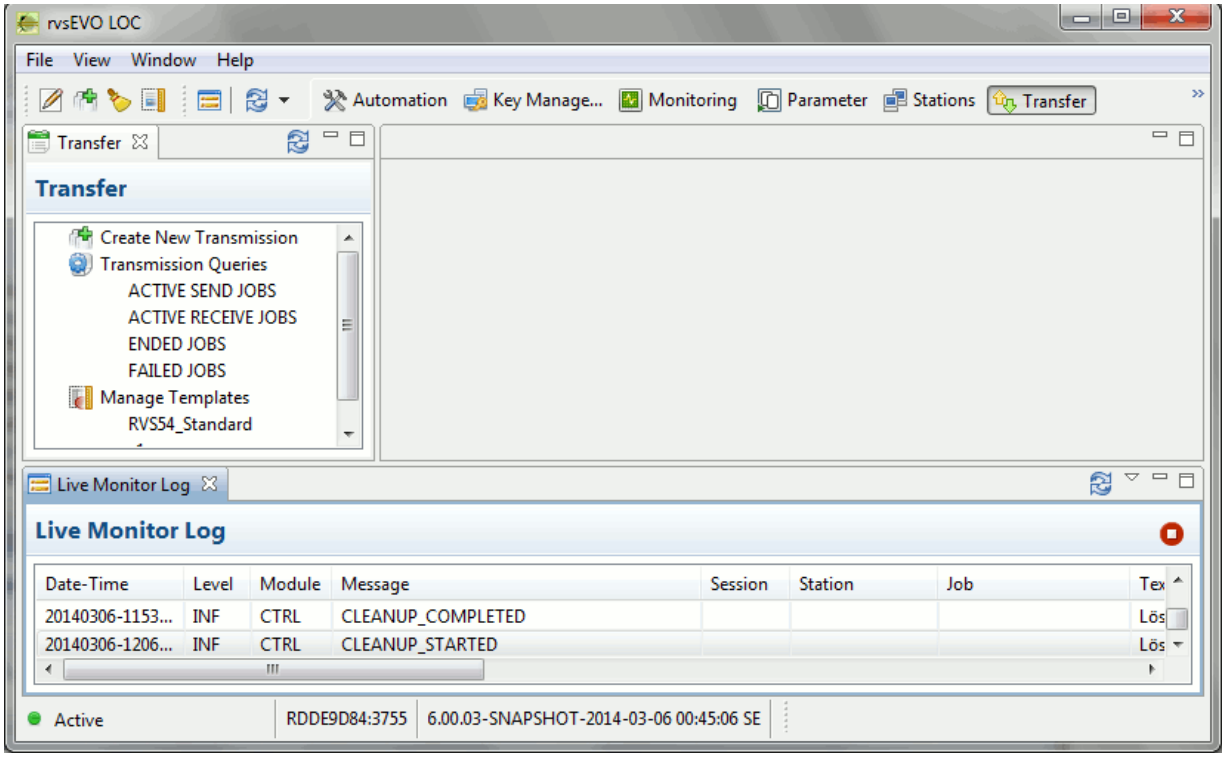
6 File Transmission with rvsEVO

This chapter provides information concerning file transmssion and output of transmission data via Transfer perspective.

6.1 User Interface of Transfer Perspective

Select the menu item `View` in the function bar and thereafter the sub-entry `Transfer` to open the Transfer perspective.

The illustration below shows the default view of Standard Edition:



The symbols in the function bar provide the following functionalities:

– Add New Query
– Create New Transmission
– Cleanup (deletion of jobdata)
– Create New Template
– Live Monitor Log
– Reset Perspective

Below, the nevigation tree provides the following entries:

– Create New Transmission
– Transmission Qeries
– Manage Templates

Via a mouse click on the small arrow to the left of `Transmission` or `Manage Tempates` entry your queries and templates can be blended or blanked.

The Live Monitor-Log View is positioned at the bottom of the GUI.

6.2 Create a Send Job

Hints:

Instruction for entering the parameter values:

If the input field includes an arrow you must select one of the specified values from the combobox.

In input fields without arrow you must directly write the value into the input field.

Exeption: Station IDs can be written into the input feld or selected from the combobox. If you enter the value dircetly into the input field you can use the wild-cards '*' and '?'.

A checkbox is to be enabled or disabled by single-click in the checkbox.

Hints for local delivery

There are two ways for sending files to the own local or virtual station.

Looptest: The file is transmitted via a neighbour station back to the own station. For this procedure **LooptestNeighbourSID** parameter has to be set. (see chapter 10.7 "ODETTE Parameters").

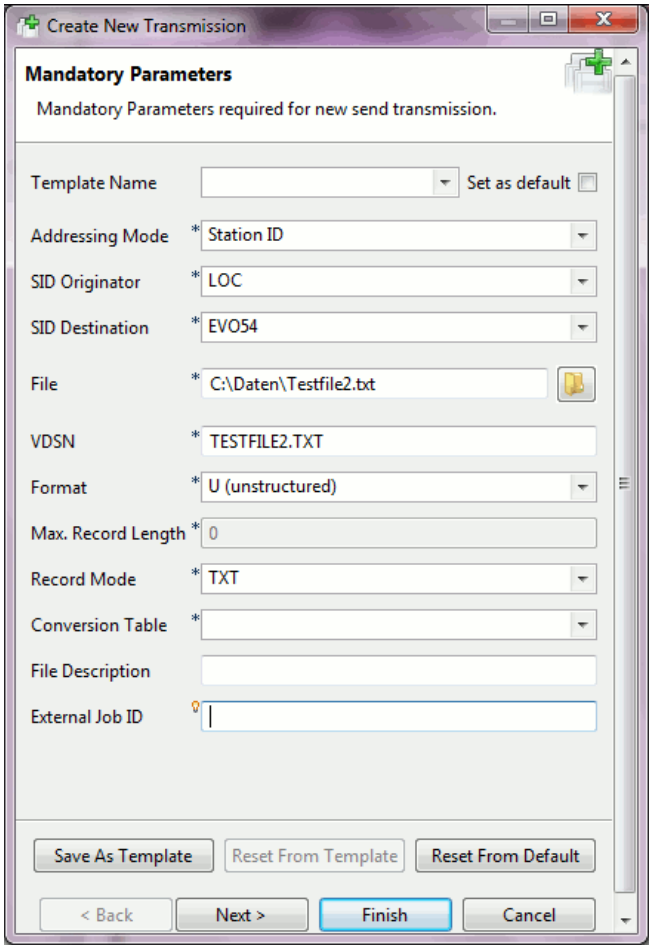
Local delivery without network connection. For this procedure **Looptest-NeighbourSID** parameter must not be set (see chapter 10.7 "ODETTE Parameters").

Follow the introduction below to create a send job.

- Press the `Create New Transmission` entry in the navigation tree or `Create New Transmission` symbol in the function bar to open the wizard for creating a send job.
- Input the mandatory parameters in the first dialog. Press **Next** to set up the optional parameters or **Finish** for starting the send job.
  Hint: In table “Required send parameters:” on page 81 you can find the description of the parameters.

Templates

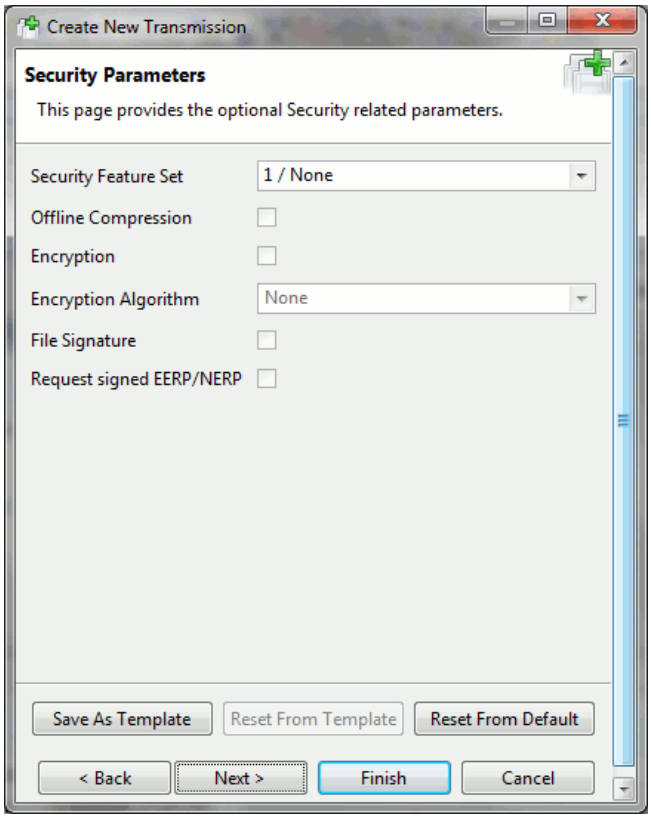
There is the possiblity to create templates for data input. (See chapter “Create, Send and Remove Templates for Send Jobs” on page 86 for further information). Press the arrow to the right of input box `Template Name` to get the list of created templates and select a template that is to be used. You can edit the imported values. The job is started via a mouse click on **Finish** button.



- In the next dialog the security parameters can be set up. Click **Finish** to start the job or **Next** to set up the general parameters.

  Hint: If you use `File Service Module` the security parameters have to set up via `Stations` perspective.

  In table “Security parameters:” on page 84 you can find the description of the parameters.

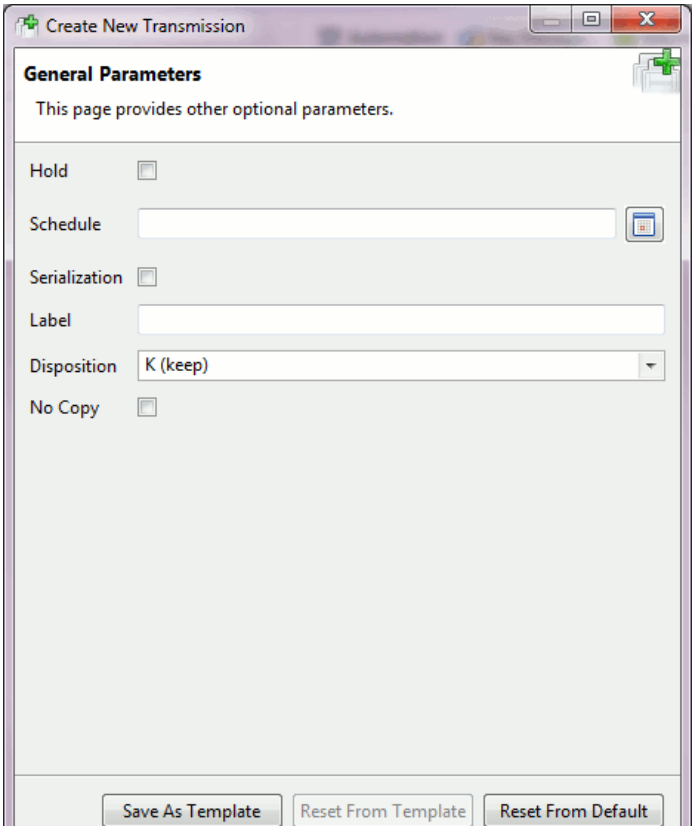


- In the following window further optional parameters can be set up. After you finished the input, press **Finish** button to start the send job.

  Hint: In table “Other Optional Parameters:” on page 85 you can find the description of the parameters.

  Before starting your send job, you have the possibility to save the job as template via mouse click on **Save as Template** button.

.



Hint: You can also use the `createSendJob` script to create a send job (see chapter 12.3.2 "createSendJob")

6.2.1 Send Parameters

The tables below give a description of all send parameters.

Required send parameters:

Parameter	Description
Template Name	This parameter offers the opportunity to select a template from the drop down menu, if they were previous created.
Adressing Mode	With this parameter you can indicate whether the sendjob shoud be created via station ID or via Odette ID. By default this parameter is not displayed and the station IDs must be inserted. With `<entry key="client.sendjobs.allow.OID.addressing" value="yes"/>` definition in `$RVSHOME/conf/rvsEvoClient.prefs` file you can set up the display of **Adressing Mode**.

Required send parameters:

Parameter	Description
SID Originator	Station ID of originator. Possible values: - SID of local station (default) - SID of virtual Station **Hint:** Depending on parameter **Adressing Mode** input of **SID Originator** or **OID Originator** is required.
SID Destination	Station ID of destination. **Hint:** Depending on parameter **Adressing Mode** input of **SID Destination** or **OID Destination** is required.
OID Originator	Odette ID of originator. possible values: - OID of local station (default) - OID of virtual station **Hint:** Depending on parameter **Adressing Mode** input of **SID Originator** or **OID Originator** is required.
OID Destination	Odette ID of Destination. **Hint:** Depending on parameter **Adressing Mode** input of **SID Destination** or **OID Destination** is required.
File	Name of the file to be send.
VDSN	Virtual file name; the length of the file name used for ODETTE transfer must not exceed 26 characters. Default: Name of the File to be transmitted. **Hint:** Via defintion `rvs_evo.controller.underscore_in_vdsn_allowed=true` in `$RVS_HOME/conf/rvs-system.properties` file you can configure that underscores are allowed in VDSN.
Format	Format of the file to be sent: - T=text file; a stream of ASCII characters - U=unstructured (binary); byte stream - V=variable; variable record length - F=fixed; fixed record length Please see also the parameter **Record Mode**.
Determine Record Length automatically	Only for files in format 'V' and if parameter **Security Feature Set** is '4': If the checkbox is activated the parameter **Max. Recorod Length** is not to be set up and the record length is determined automatically.
Max. Record Length	Maximal record length for the files in format 'F' or 'V', please see also the parameter **Record Mode**

Required send parameters:

Parameter	Description
Record Mode	In this parameter you indicate the mode of files in format 'F' or 'V'. Possible values: - TXT: text file - BIN: binary file Text File in format 'F': If the record length deviate from the specified one (Parameter **MaxRecl**) the line will be filled with blanks or will be cut up to defined value. Text File in format 'V': longer records will be cut up to defined value. Binary files: The length of all records is the max. record size except the length of the last one. Binary files in format 'F': If the size of the last record is shorter than the specified one the line will be filled to the defined value.
Conversion Table	For ASCII - EBCDIC conversion the following conversion tables are available: ASCII-IBM037, ASCII-IBM273, ANSI-IBM073, ANSI-IBM273. For EBCDIC - ASCII conversion the following conversion tables are available: IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI. For using your own conversion table, please read the note for "Conversion tables" on page 133.
External Job ID	Parameter for an external JobID, which can refer to several rvsEVO JobIDs. The external JobID is a string of alphanumeric characters. By default, **External Job ID** is an optional parameter. With `<entry key="client.transfer.sendJobWizard.externalJobId" value="mandatory"/>` definition in `$RVSHOME/conf/rvsEvoClient.prefs` file you can set up **External Job ID** to mandatory parameter. This setting is only valid for sendjobs what are created via rvsEVO GUI.
File Description	Comment, free text, available only in connection with OFTP version 2. If your partner does not support OFTP version 2, this field will be ignored.

Security parameters:

Parameter	Description
Security Feature Set	This parameter applies to the format of encryption. According to the selected kind of the encryption, the groups of available parameters will be activated. Following values are possible: - `1 / None`: no encryption - `2 / ComSecure (V1)` - `3 / ComSecure (V2)` - `4 / OFTP 2.0 (CMS)` rvsEVO supports two formats of encryption: ComSecure (an appropriate format) und CMS (based on X.509 certificates). CMS format is only in with OFTP version 2 possible (your partner has to support OFTP version 2.0, too). ComSecure (V1) means version 1 (supported by the ComSecure product versions 1.1 and 1.2). ComSecure (V2) means version 2 (supported by the ComSecure product version 1.3). V2 is needed for the encryption of the files bigger than 4 Gbyte. Your partner must support this ComSecure version, too.
Offline Compression	Compression; active only in connection with parameter **Security Feature Set**=2, 3 oder 4. If the checkbox is activated Odette compression takes place.
Encryption	File encryption; active only in connection with **Security Feature Set**=2, 3 oder 4; If the checkbox is activated encryption takes place.
Encryption Algorithm	Active only in connection with OFTP 2.0 (CMS) (see parameter **Security Feature Set** in this table. The following algorithm are possible: - `None` - `DES_DED3_CBC` (Triple DES) - `AES256_CBC`
File Signature	Active only in connection with OFTP 2.0 (CMS) (see parameter **Security Feature Set** in this table. If the checkbox is activated file signature takes place

Security parameters:

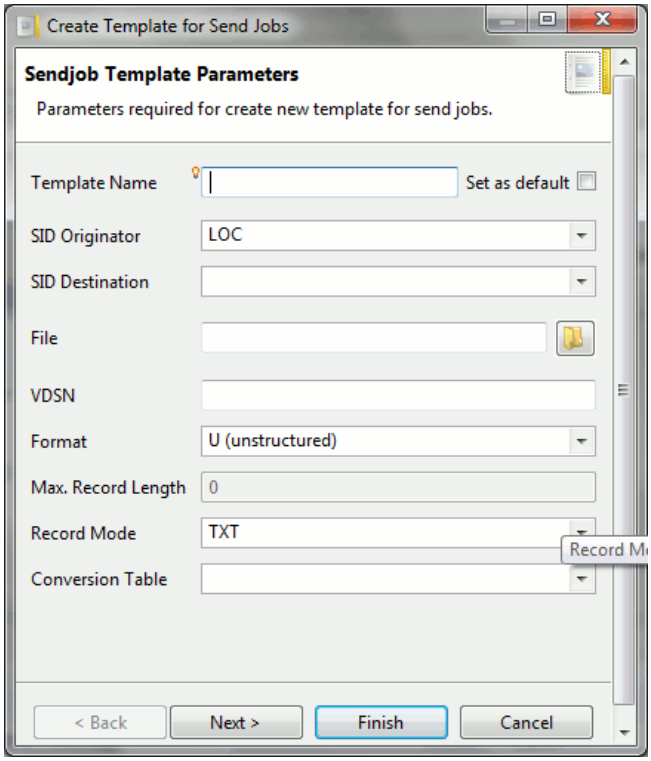
Parameter	Description
Request signed EERP/ NERP	Active only in connection with OFTP 2.0 (CMS) (see parameter **Security Feature Set** in this table). In OFTP 2.0 there is a possibility to request the signature of the EERP or of the NERP. Your partner must support OFTP 2.0 in this case. If the checkbox is activated signed EERP or NERP is requested and the sendjob ends in FAILED status if your partner sends no signed EERP or NERP.

Other Optional Parameters:

Parameter	Description
Hold	If the checkbox is activated the send job is created in HOLD status.
Schedule	With this parameter you can define what time the SendJob should be launched. Press the icon on the right to select date and time. Default: at once.
Serialization	If the checkbox is activated the files will be sent in the same order, as the send jobs were created. The next job will only be sent, if the previous is completely finished. All send jobs for the serialization must have the same label (see parameter **Label**).
Label	Group name of serialized send jobs. User specified (descriptive) label for this job. (See parameter **Serialization**)
Disposition	Disposition of local file after successful send attempt. Possible values: - K (Keep): file will not be deleted after sending - D (Delete): file will be deleted after sending.
No Copy	With this parameter you can decide whether the file which should be transmitted is to copy to Outbox directory or not. **Attention**: If you have installed a serverfarm, this parameter must not be activated. The transmitted files need to be copied to outbox directory on the NFS drive, so that each node is able to access the file. (see also chapter 15 "Multi Instance Functionality of rvsEVO"). **Hint**: no effect if files in U format are sent with code conversion

6.3 Create, Send and Remove Templates for Send Jobs

Right-click `Manage Template` entry in the navigation tree and thereafter `Create New Template` to create a template or press `Create New Template` icon in the function bar. In the dialog below you can configure the mandatory parameters:



Feed the name of the template in input field `Template Name`. Your template will be saved in **`Manage Template`** entry of navigation tree.

If a template should be used by default, activate the `Default` checkbox.

Hint: Please see chapter 6.2.1 "Send Parameters" for the description of the other parameters. The procedure of creating a template is analogue to creating a send job.

Edit template

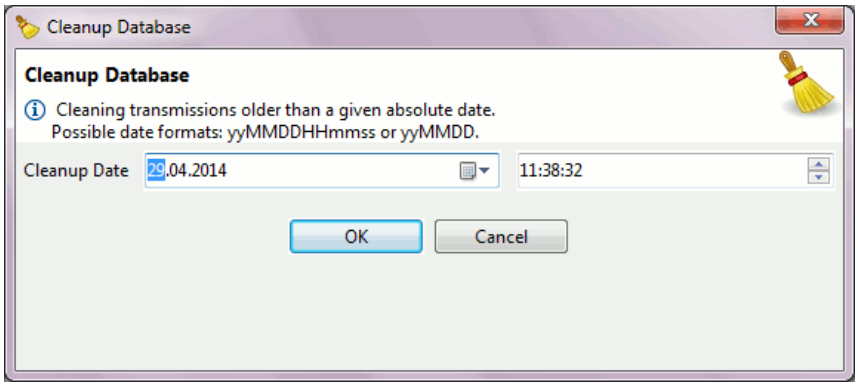
Right-click the template to be changed and thereafter select the entry `Edit Template` for changing a template. The window `Update Template for Send Job` is opened and the parameters can be edited. Press **Finish** button to save the template.

Delete template

Right-click the template to be removed and thereafter select the entry `Delete Template` for removing a template. Press **Yes** Button to delete the template.

6.4 Deletion of Job Data from Database

Click the **Cleanup** icon in the toolbar for deletion of ended and failed jobs from the database. The jobs which are older than the entered date and time will be deleted.



6.5 Display of Receive and Send Jobs

Listings and detailed information of jobs can be output via `Transmission Queries` entry in the navigation tree:

Job lists

The following default queries are provided by rvsEVO:

– `ACTIVE SEND JOBS`: listing of active send jobs

– `ACTIVE RECEIVE JOBS`: listing of active receive jobs

– `ENDED JOBS`: listing of ended jobs

– `FAILED JOBS`: listing of failed jobs

They can be opened with a right-click on the query and then selecting `Open Definition.` Right-click the query and select `Open Definition in New View` to open the query in a new view. The view in foreground is to be selected via tab.

A query can be executed with a simple click on the query or via the **Execute** button in the opened view. Alternatively you can right-click the query and afterwards select `Open Query Result in New View`. In chapter 6.5.1 "Create Transmission Queries" you come to know how to generate your own transmission queries.

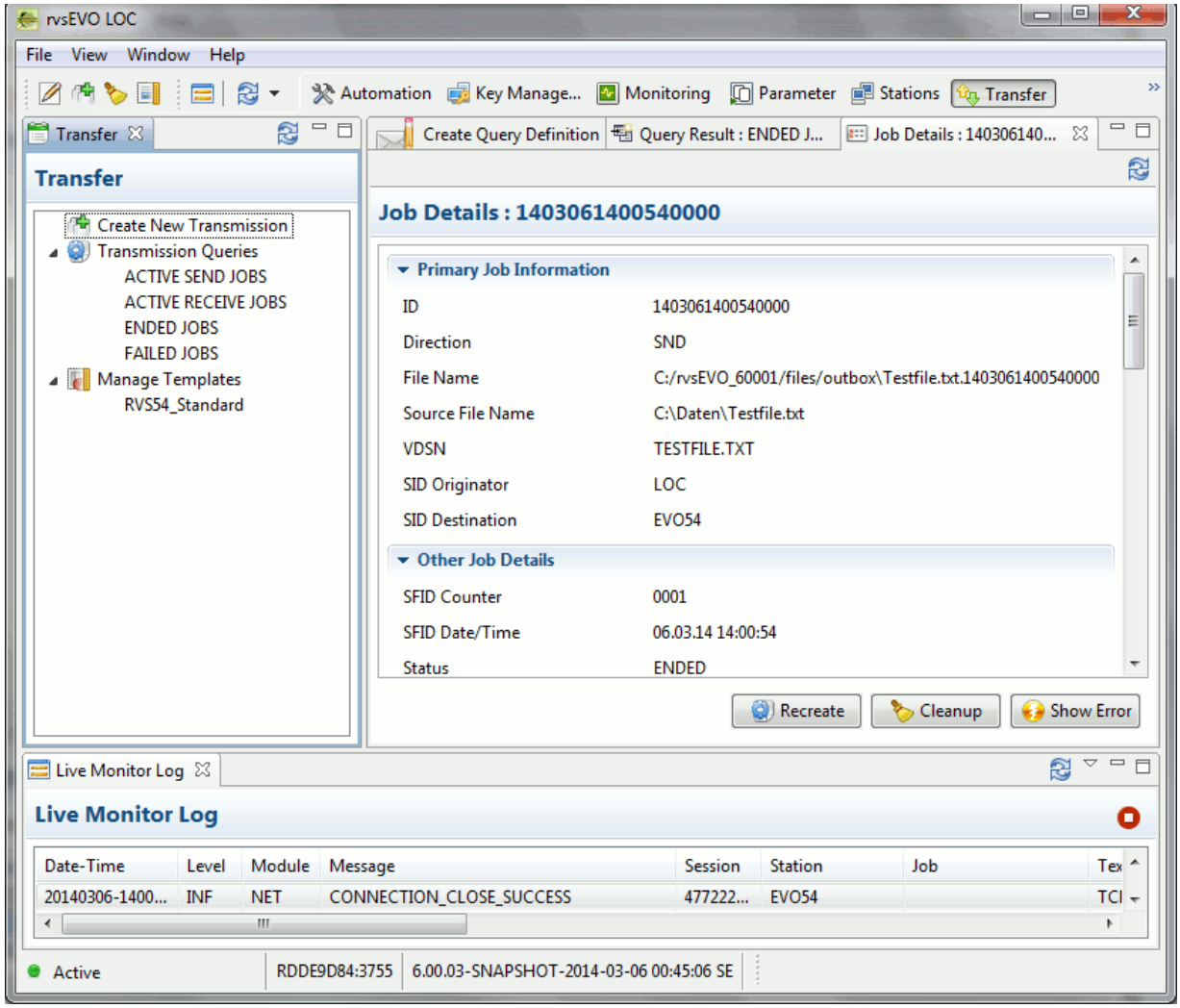
Create Defatult Queries

If the queries are changed or removed you have the possibility to generate new ones. This functionality is executable via the tool `$RVS_HOME/tools/createDefaultQueries`.

Detailed view

With a double-click on the job in the list you get the detailed view of the transmission job. Another way to output the job details is to right-click the job and thereafter select `Open in New View`. In the second case the detailed view of several jobs can be opened . The view in foreground is to be selected via tab.

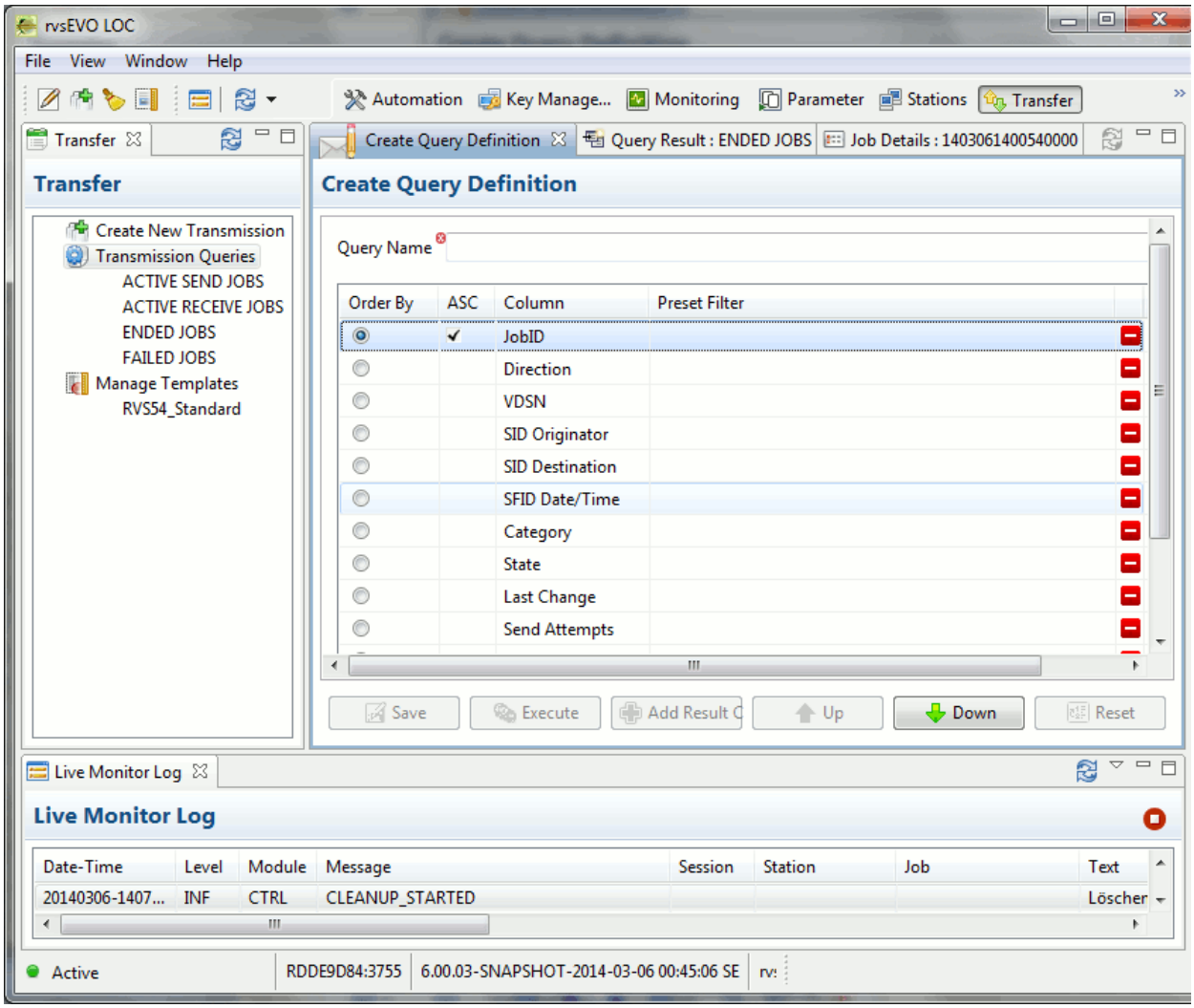
The illustration below shows the `Transfer` perspective with the job list at the bottom and the detailed view on the right-hand area:



6.5.1 Create Transmission Queries

To create transmission queries proceed as follows:

- Right-click `Transmission Queries` entry in the navigation tree and select `Add New Query` from drop-down menu or press the `Add New Query` symbol in the function bar to open the wizard for creating a transmission query. The following window opens:



- Feed the name of the query in the input field `Query Name`. Your query will be listened in **`Transmission Queries`** entry of navigation tree.
- Feed the needed filter criteria in column `Preset Filter`. (e.g. with parameter **Direction** 'SND' for send job or 'RCV' for receive job)
- Activate the radio button in `Order By` column next to that parameter if you want to sort the output.
- Use the **Add** (**+**) and **Delete** (**-**) buttons on the right-hand side for adding or removing criteria.
- Save the query via **Save** button after configuring the needed criteras.

Hint:

You can use the filter criterion `Last Change` with wildcards ('?' as wildcard for one sign and '*' for multiple signs) and '<' and '>' signs. The dateformat must be `YYYY-MM-DD hh:mm:ss`.

Example:

`< 2013-04-25 10:00:17`: Jobs older than the given date.

From rvsEVO version 6.01.03. onwards filters of type relative date can be set. You can use the operators „older than“ and „younger than“ with the following time units: „minute“, „minutes“, „hour“, „hours“, „day“, „days“, „month“, „months“, „year“ and „years“

Examples:

`older than 3 days`

`younger than 1 month`

Use „from“ ... „until“ for filtering jobs of a special period.

Example:

`from 2015-10-01* until 2015-10-31*`

Press `Save` button for saving your changes.

Press the `Reset` button for deleting your inputs.

Click the `Execute` button to execute the query.

6.5.2 Editing of Transmission Queries

Open the detailed view of the transmission query to be changed with a right-click on this query and then select `Open Definition` or `Open Definition in New View` from the context menu. In this view the filter criterias can be added, deleted or changed like described in chapter 6.5.1 "Create Transmission Queries".

6.5.3 Deleting of Transmission Queries

You can delete a query via a right-click on the query and then selecting `Delete Query` from the drop-down menu.

6.5.4 Job parameters

In the following tables the job attributes are described.

Primary Job Information:

Parameter	Description
ID	rvsEVO job ID
Direction	Direction: SND=sending or RCV=receiving
VDSN	Virtual file name for Odette-transfer.
SID Originator	SID of the Originator station
SID Destination	SID of the recipient

Primary Job Information:

Parameter	Description
Source File Name	The entire path of the original file; this file is copied in the rvsEVO outbox directory.
Filename	Path and name of the received or sent file (Default is the file in `inbox` or `outbox` directory).
File Description	File-description. Text comment, possible with OFTP version 2. If your destination device is not OFTP 2.0, this field is ignored.
External Job ID	ID for external application

Other Job Details

Parameter	Description
Status	Job status. For possible status values, please see the rvsEVO-user manual 6.5.5 "Status for Send Jobs and Receive Jobs"
Last Change	Time of the last change in the job status
SFID Date/Time	Date and time of the creation of the transmission job
SFID Counter	Sequential counter from SFID, if several send jobs were created at the same time. (beginning with 0001)
Schedule Time	Fixed date for the transmission
Time Start File	Restart position, in case of which a new transfer of the file should be started (system time in milliseconds)
Time of first Send Attempt	Date and time of the first attempt for a connection with your neighbour station.
Retry Count	Number of failed send attempts.
Send Attempts	Number of faulty dispatch attempts. After successful transmission the parameter is set to zero.
Bytes Transmitted	Number of transmitted bytes
Restart Position	Restart position, in case of which a new transfer of the file should be started

File Features

Parameter	Description
Size of Source File	Size of original file before processing through the service provider
Format	Format of the file to be sent: - T=text file; a stream of ASCII characters - U=unstructured (binary); byte stream - V (variable); variable record length - F (fixed); fixed record length. Please see also the parameter **Record Mode**.
Record Length	Maximum record length
Record Mode	In this parameter you indicate the mode of files in format 'F' or 'V'. Possible values: - TXT: text file - BIN: binary file
Code Table	Name and path to conversion table
Disposition	This parameter decides whether the file to be sent remains as it is, or is deleted locally after sending. Possible values: - K (Keep): File remains as it is, after sending - D (Delete): File is deleted after sending
Serialization	If this parameter is activated your files are send in a serial order. All the files which should be sent in the same group, should have the same ID (Label).
Label	Label for serialization. All the files which have been sent in the same group, should have the same ID (label). See also parameter **Serialization**

Security Features

Parameter	Description
Security Feature Set	Security Features; possible values: - 1 (none) - 2 (ComSecure V1) - 3 (ComSecure V2) - 4 (OFTP 2.0)
Compression	Odette compressing, if the checkbox is activated.
Encryption	Encoding, if the checkbox is activated

Security Features

Parameter	Description
Encryption Algorithm	Encoding algorithm; possible values: - `blank` (no algorithm) - `DES_EDE3_CBC (Triple DES)` - `AES256_CBC`
File Signature	If the checkbox is activated, the file is signed
Request Signed EERP/ NERP	If the checkbox is activated, signed EERP/NERP is requested
NERP Reason Code	Code of NERP
NERP Reason Text	Description of NERP Codes
NERP Creator OdetteID	Odette-ID of the creator of the NERP
Encryption Cert. Serial Number	Serial number of the certificate. A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
Encryption Cert. Issuer Name	Publisher name of the certificate. A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
Sign Cert. Serial Number	Serial number of the signature A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
Sign Cert. Issuer Name	Publisher name of the signature A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.

Job Error Information

Parameter	Description
Error Number	Error code
Description	Description of error code

6.5.5 Status for Send Jobs and Receive Jobs

Status for Send Jobs and Receive Jobs:

– `RESTART` (wait after interruption of transmission to send file again)

– `SP_PROCESSING` (Service provider is processing job)

- SP_ENDED (Service provider has processed job successfully)
- SP_FAILED (Service provider failed during processing of job)
- ENDED (Job has successfully ended)
- ENDED_WITH_JS_ERROR (Job run into error during call of jobstart)
- FATAL_ERROR (Job run into fatal error)
- DELETED (Job got deleted)

Status for Send Jobs:

- CREATED (Job was created, but not sent)
- WF_SFID_ANSWER (Sendjob waits for answer to already sended SFID)
- WF_CDT (Sendjob waits for credit renewal)
- WF_EFID_ANSWER (Sendjob waits for answer to already sended EFID)
- WF_EERP (Sendjob waits for EERP)
- WF_EERP_ROUTING (Wait for EERP during Routing)
- SYNCHRONIZE_ERROR (Sendjob run into synchronize error)
- HELD (Sendjob got holded)
- RELEASED (Sendjob got released)
- FAILED_WITH_NERP (NERP received)
- FAILED_WITH_SFNA (Receiving of SFNA with setting Restart = No)
- FAILED_WITH_INVALID_PROTOCOL_VERSION (Transmissions failed based on invalid protocol version)
- SCHEDULED (Send job created - execution deadline not reached)

Status for Receive Jobs:

- RESTART_AFTER_EFNA (Transmission with error; EFNA send; waiting for retransmission)
- RESTART_AFTER_EFPA_FAILURE (File already delivered but EFPA failed; assuming partner will send file again)
- RECEIVING (Receiving file data after EFPA)
- EERP_HELD (File was completely received; EERP is on hold and needs to get released)
- EERP_RELEASED (User released EERP, but it is still not send)
- EERP_DELETED (User deleted EERP; job will stop)
- NERP_RELEASED_JS (NERP released by JS, but it is still not send)
- FAILED_WITHOUT_NERP (transmission failed; no NERP sended)
- ROUTING_HELD_DURING_SHUTDOWN (Couldn't create routing send job, because of shutdown)

– FAILED_WITH_INVALID_EERP_SIGNATUR (Received EERP with invalid signature)

– FAILED_WITH_INVALID_NERP_SIGNATUR (Received NERP with invalid signature)

Hint: Please see chapter 16 "ODETTE Protocol" for ODETTE protocol sequences (such as EFNA, SFID, EFID, EERP, ...).

6.6 Operations on Jobs

You have the possibility to execute activities for the job via a right-click on the job in the list or via the buttons at the bottom in the detailed view:

Active Send Transmissions

– Open In New View

– Copy (to copy one or more jobs from the job list and insert them into a text file)

– Forced Delete (allows to delete an active job without pausing it before)

– Restart

– Delete Transmission

– Release Transmission

– Hold Transmission

Active Receive Transmission

– Open In New View

– Copy (to copy one or more jobs from the job list and insert them into a text file)

– Restart SP (after bug fixing, to activate a job in SP_FAILED status)

– Delete EERP

– Forced Delete (allows to delete an active job without pausing it before)

– Release EERP

– Release NERP

Ended Transmissions

– Open in New View

– Copy (to copy one or more jobs from the job list and insert them into a text file)

– Recreate (only for ended send jobs)

– Cleanup (deletion of transmission data from the database)

Failed Transmissions

– Open in New View

– Copy (to copy one or more jobs from the job list and insert them into a text file)

- Recreate (only for send jobs)
- Cleanup (deletion of transmission data from the database)

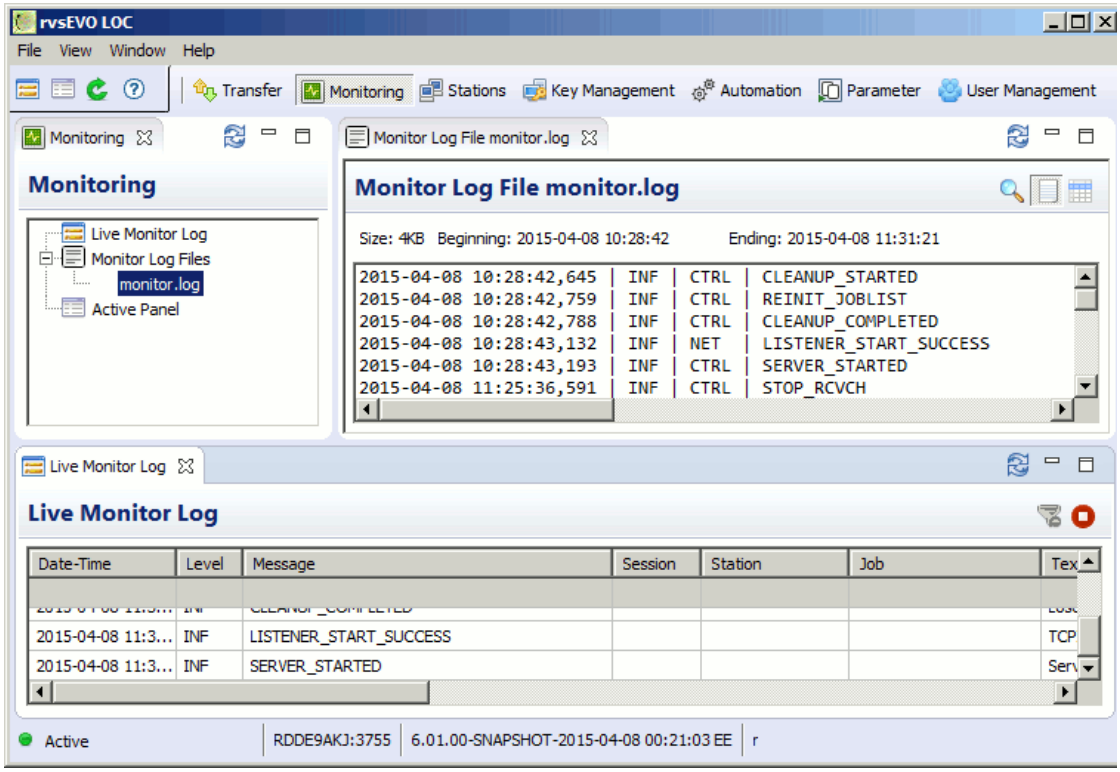
7 Monitoring

In this chapter you find out how to output rvsEVO messages and warnings.

7.1 User Interface of Monitoring Perspective

With default view Log Messages are normally displayed in the bottom of the GUI window.

You can view new and old messages via `Monitoring` Perspective. The `Monitoring` perspective is reachable via menu item `View -> Stations`. The illustration below shows the default view of `Monitoring` perspective in Standard Edition:



The symbols in the function bar provide the following functionalities:

– Cleanup MonLog

– Live Monitor Log

– Reset Perspective.

Below the navigation tree of `Monitor Tree View` contains the following entries:

– Live Monitor Log (current Log messages)

– Monitor Log Queries (only with Enterprise Edition and Oracle- or MS SQL database)

– Monitor Log Files (old Log messages)

– Active Panel

7.2 Live Monitor Log

Via a single-click on the `Live Monitor Log` entry in the navigation tree the current messages are displayed at the bottom of the window in a new view.

Language Setting

The language setting of the `Live Monitor Log` can be aligned via `user.language` parameter in `$RVS_HOME/conf/rvs.properties` file. Changes only take effect after a restart of rvsEVO server.

Configuration

The latest 1000 messages are displayed by default. This setting can be configured via the definition `<entry key="client.monitorlog.livelog.pagesize" value="1000"/>` in `$RVS_HOME/conf/rvsEvoClient.prefs` file.

Anymore it is possible to highlight the display of information (`INF`), warnings (`WRN`) and error messages (`ERR`). This is configurable via the definitions `<entry key="client.monitorlog.[INF|WRN|ERR].colour.[r|g|b]" value="[255|0]"/>` in `$RVS_HOME/conf/rvsEvoClient.prefs` file.

Examples:

```
<entry key="client.monitorlog.ERR.colour.r"
value="255"/>
```

In the example above the error messages are displayed red (`colour.r`).

```
<entry key="client.monitorlog.WRN.colour.g"
value="255"/>
```

In the example above warnings are displayed in green (`colour.g`).

Hint: The GUI must be restarted to make the changes effective.

By default the `Live Monitor Log` is automatically updated. Via the button  the auto refresh can be stopped and via the button  the auto refresh can be started again.

Hint:

Via the buttons  at the top of the `Live Monitor Log View` you have the possibility to refresh the `Live Monitor Log View`, to select the columns to be displayed and to minimize or maximize the window.

The following information are displayed:

Monitor-Log parameters

Parameter	Description
Date-Time	Date and time of the entry (format `yyyyMMdd-HHmmss`)

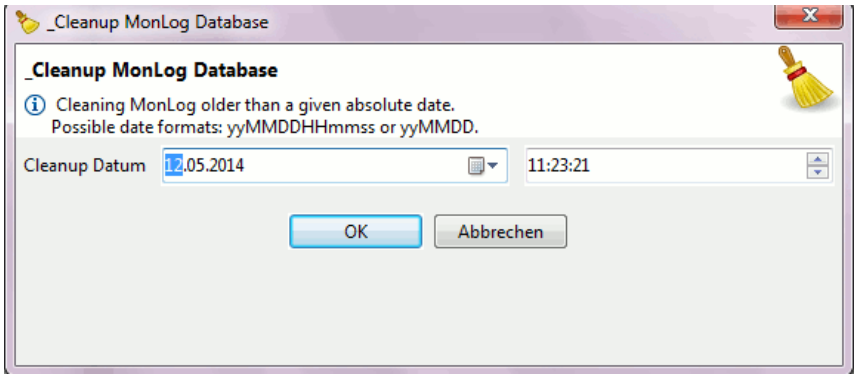
Monitor-Log parameters

Parameter	Description
Level	Kind of message. Possible values: - INF: Information - WRN: Warning / wichtiger Hinweis - ERR: ERROR
Instance	Node Name (only important for Muli Instance Functionality)
Module	rvsEVO-Modul (sender of the message) Possible values: - SRV: rvsEVO server - CTRL: controller - NET: network - OFTP: Odette protocol - CONF: configration - PERS: persistence - ACX: automatic exchange of certificates - SP: Service Provider - OBSR: Observer
Message	Message key
Session	ID of OFTP session
Station	SID of neighbour station
Job	rvsEVO JobID
Text	Text message from message key (GUI) or parameter for text message (Log file)

7.3 Deletion of MonLog Entries from Database

If adjustment of MAINTENANCE-Parameter **MonlogInDatabase** = Y the database table must be cleaned up regularly because otherwise it would fill up.

Single-click the `Cleanup Monlog` icon in the toolbar to open the dialog window for deletion of MonLog data. All entries are deleted wich are older than defined date.

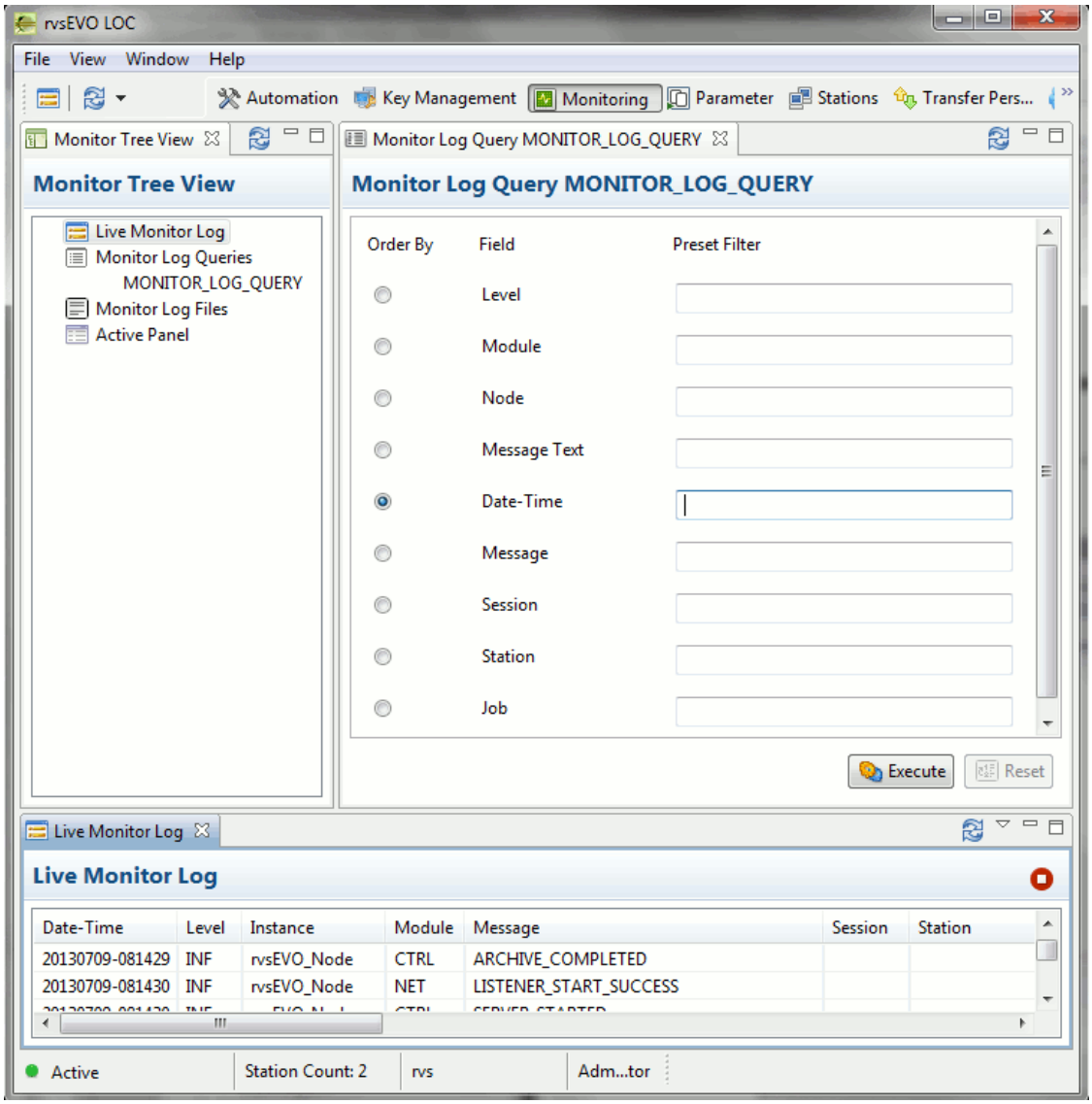


You can also execute this functionality via `clenaup` commandline tool.

7.4 Monitor Log Queries

Hint: This functionality is only available for rvsEVO Enterprise Edition if using an Oracle- or MS SQL database.

Monitor Log queries can be configured via `Monitor Log Queries` entry in the `Monitor Tree View`. Via a single-click on the arrow on the right next to the `Monitor Log Query` entry and selection of `MONITOR_LOG_QUERY` from the drop-down menu the query view opens:



7.4.1 Define Filter Criteria

To define the filter criteria proceed as follows:

- Feed the filter criteria in the input filed in column `Preset Filter`.
- Activate the radio button in `Order By` column next to that parameter you want to sort on the output.
- Execute the query via a single-click on **Execute** button.

Please see table “Monitor-Log parameters” on page 98 to get information about the parameters.

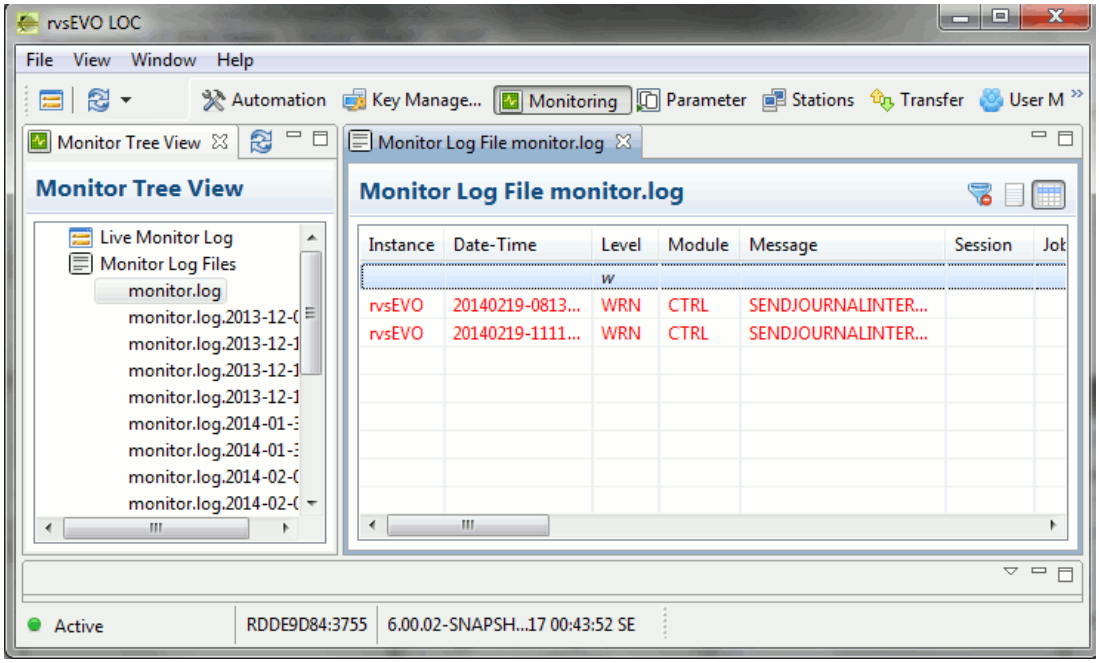
7.5 Monitor Log Files

The current and the old messages are saved in the log files and can be displayed via a single-click on `Monitor Log Files` entry in the `Monitor Tree View`. A mouse-click on a log file displays the detailed information of the log file in a new view.

Via **Text View** and **List View** symbols at the top of the view the output format can be selected.

List view

The list view provides a filter function: Insert the string you want to filter on into the correspondent column. (E.g. if you insert „`w`“ in column `Level` only the warnings are displayed“.) You can use the „`*`“ wildcard. (E.g. if you insert „`*r`“ in column `Level` the warnings (WRN) and Errors (ERR) are displayed“.) Click the **Reset Filter** symbol for resetting the filter.



Text View

The text view provides the text retrieval. You can open the search assistent via the **Find** symbol (loupe) at the top of the view or via hot key **Ctrl+F**.

Via hot key **Ctrl+A** the whole list can be selected and via hot key **Ctrl+C** you can copy the selected section into the clipboard. This functions are also executable via a right-click in the view and subsequent selection of `Select All` respectively `Copy` from the drop down menu.

Hint: Please see chapter 12.12 "Display Monitor Messages" in order to find out how to display log messages via commad line tool.

7.6 Active Panel

This tool gives you the additional details of the status of an active transfer.

A mouse-click on the `Active Panel` entry in the `Monitor Tree View` opens a list of current transfers in a new view.

The following information is available:

Active Panel parameters

Parameter	Description
Neighbour	Station ID of the neighbour station
State	`Initiator` (own station is an active side of the communication process); `Responder` (own station is the passive side of the communication process)
Command	`Create Session`: creation of session `Start File`: beginning of the transfer `Transmit File`: transfer of the file `End File`: end of the transfer
Originator	SID of originator
Destination	SID of destination
Direction	`Sending` `Receiving`
File Name	virtual file name
Line Speed	transfer rate in kB/sec
Progress	percentage closed content of the transfer
Start File	Time of the beginning of the transfer.
SessionID	Is automatically signed for each transfer
Start Session	Time of the session start
Line Type	Network type

Hints:

You can get a sorted output by clicking on the parameter (column headings) you want to sort.

Also a filter function is provided. Please read chapter 7.5 "Monitor Log Files" section "List view" for description of the filter function.

Use the symbol for refreshing at the upper right-hand border for refreshing the list.

Terminate Session

Via the **Terminate Session** button a session can be terminated. There is also a symbol at the upper right-hand for this functionality.

8 Secure Transmission with rvsEVO

Basics

For secure transmission risks such as the attack on the confidentiality, intergrity and authentication of your data must be excluded. Also the traceability of the transmission must be ensured. In rvsEVO this is achieved with the following features:

- line encryption via TSL protocol
- OFTP2 authentication via AES or 3DES and RSA
- file encryption via AES or 3DES and RSA
- file signature
- signature of EERP/NERP

8.1 Line Encryption

Line Encryption is the encryption of the network connection between two direct neighbour stations during the transmission. In rvsEVO the TLS protocol (**T**ransport **L**ayer **S**ecurity) is used therefore.

8.2 File Encryption

The file to be send is encrypted by the sender and decrypted by the receiver.

rvsEVO offers two file encryption formats: **ComSecure** and **CMS**.

Com-Secure is an appropriate format, which is used for rvs® portable, too.

CMS (Cryptographic Message Syntax) is an Internet Standard (rfc 2630) for the syntax of an encrypted message. This standard is implemented in the OFTP version 2.

The encryption in rvsEVO (for both formats: ComSecure and CMS) combines the benefits of symmetrical and asymmetrical techniques: the high speed of the symmetrical and the security level of the asymmetrical technique. rvsEVO uses the following techniques:

- **3DES** for ComSecure as symmetrical technique (length: 3x56 bits = 168 bits)
- **3DES** and **AES** as symmetrical technique for CMS
- **RSA** as asymmetrical technique for ComSecure and CMS (length: 768 to 2048 bits).

Hint: For encrypted file transmission JCE (Java Cryptography Extension) is needed. If you try to transfer encrypted files and JCE is not installed, you will get the following message: „invalid key length“. With Windows and Linux systems the JCE is installed automatically during installation. Please see also chapter 2.3.2 "Installation on Linux and UNIX Systems" or the readme file in `$RVS_HOME/docu/` for more information.

.

8.2.1 Principle and Sequence of rvsEVO Encryption

The principles of the encryption, which will be explained in the following part, are the same for all encryption formats, mentioned in this chapter.

Distributing the public key / safely storing the privat key

Each participant in encrypted communication creates a key pair, comprising the **public key** and the **private key**.

Each participant provides the public key to each partner he expects files from. This allows the data to be exactly encrypted for the partner who sent this public key. You can safely distribute the public key since this key alone is not sufficient for decryption.

Each participant keeps his private key and stores it safely.

Three keys are required for decryption (the own key pair and the partner's public key). It will no longer be possible to decrypt files sent by the partner if one of the three necessary keys is lost. The private key is also needed for creating a signature.

8.3 Electronic Signature

rvsEVO uses electronic signature to increase security. The signature ensures that data do not undergo any unnoticed changes during transmission. The electronic signature of the End-To-End-Response (EERP / NERP) ensures the traceability of data transmission.

8.4 Key Management in rvsEVO

This chapter describes how to create, export and import keys, as well as you get information about using CA certificates..

Hint: From rvsEVO version 6.00 onwards there are fundamental improvements in the key management:

– simplified procedure for creating own key pairs, independend of the functionality.

– functional assignment of the keys / certificates via management of stations

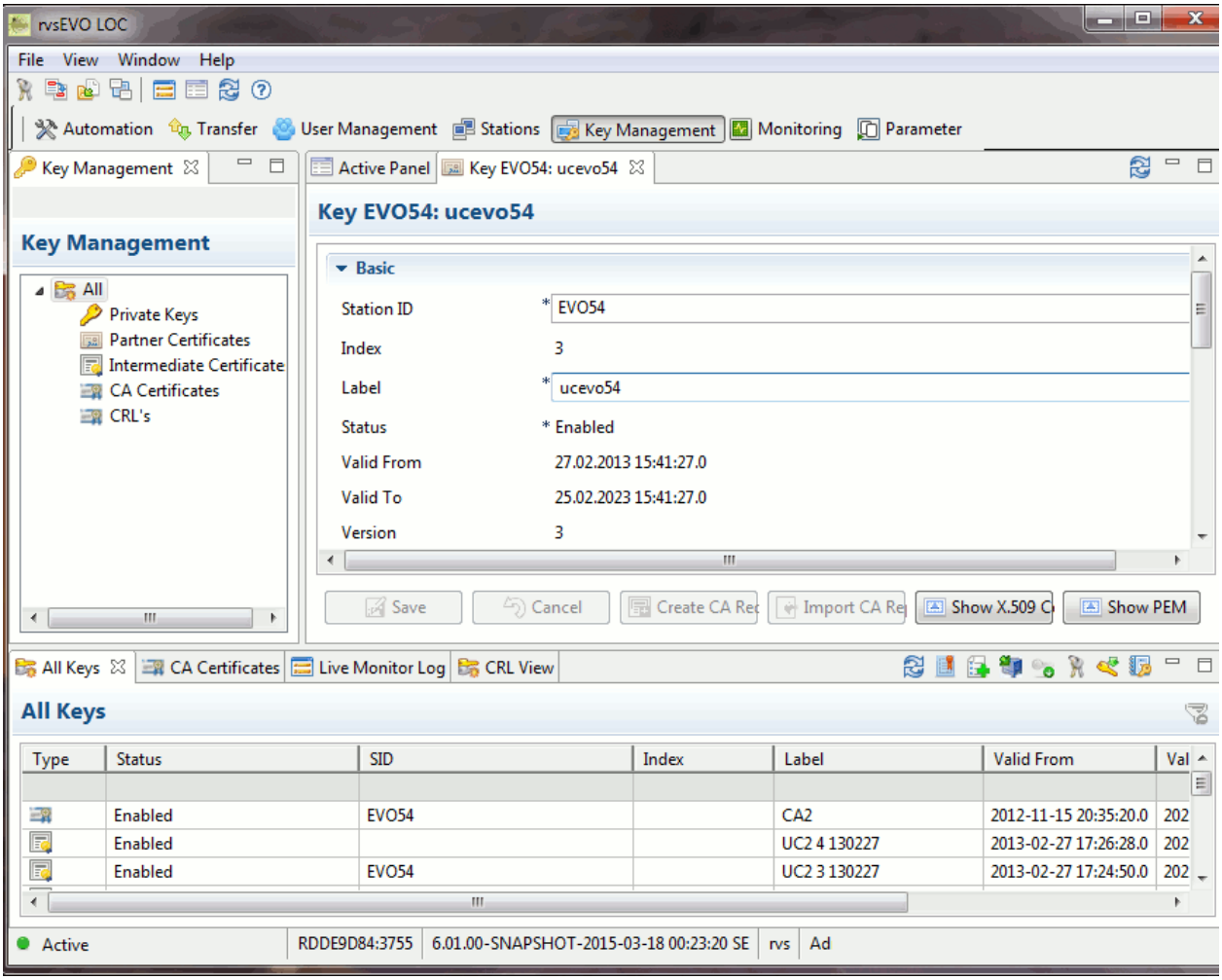
– saving of the keys / certificates in the database

– ACX Request, ACX Deliver and ACX Replace via graphical user interface.

– Only the administrator may execute the management of private keys.

8.4.1 User Interface of the Perspective of Key Management

The key management is reachable via menu item `View -> Key Management`. The illustration below shows the default view of perspective of key management in Standard Edition:



In the center/left section of the GUI window the `Key Management` view offers the following entries:

– All Certificates
  – Private Keys
  – Partner Certificates
  – Intermediate Certificates
  – CA Certificates
  – CRL's

In the center/right section the detailed view of certificates and CRL's and the list view of the „Active Panel“ is positioned. Please see chapter 7.6 "Active Panel" for more information about Active Panel.

The symbols in the function bar provide the following functionalities:

– Create Private Key
– ACX Request Certificate

– ACX Deliver Certificate
– ACX Replace Certificate
– Live Monitor Log
– Show Active Panel
– Show Reset Perspective
– rvsEVO User Manual

At the bottom of the GUI the list views of certificates and CRL's and the `Live Monitor-Log` View is displayed.

8.4.2 Necessary Steps for Encrypted Transmission via rvsEVO:

- Create your own key pair (see chapter 8.4.4)
- Export your own public key in form of X.509 certificate (see chapter 8.4.6)
- Send the own public key to your partner (via email or rvsEVO)
- The partner has to import your X.509 certificate (public key) and send his X.509 certificate to you.
- You also have to import the partners' X.509 certificate into the key management of rvsEVO (see chapter 8.4.7)
- Associate your own key via Stations Management (detailed view of local station -> parameter group **Keys**).
- Associate the certificate of your partner via the parameter group **Keys** of the partner station)

Hint for TLS encryption: The private key can only be associated if a TLS listener was created (see chapter 5.2.2.)

In chapter 6.2 "Create a Send Job" you get information about transmitting encrypted files.

8.4.3 Display Certificates

List view

Lists of certificates can be shown via a single-click on the appropriate entry in the navigation tree. You get the following information: Type, Status, SID, Index, Label and Valid period.

With a single-click on the column headings you get a sorted output. The list can be refreshed via refresh icon at the top right edge.

Via greyed out row beneath the column header you can set output filters. The wildcards '*' (for one or more signs) and '?' (for one sign) are permitted. Via 'Reset Filter' symbol you can switch off the filter function.

Hint:

Index is a unique identifier which is assigned automatically by rvsEVO.

You can copy one or more lines of the certificate list and insert them into a text file.

Detailed view

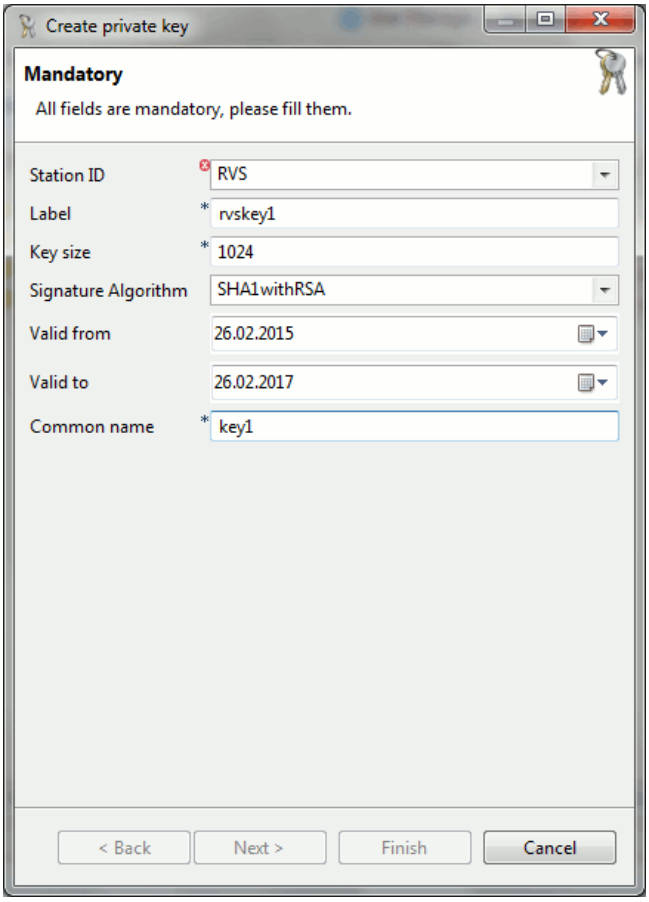
With a double-click on an entry in the certificate list you get the detailed view of the certificate. Another way to output the certificate details is to right-click the certificate in the list and thereafter select `Open in New View`. In the second case the detailed view of several certificates can be opened. The view in foreground is to be selected via tab.

In detailed view the parameters **Label** and **Station ID** can be changed.

8.4.4 Generate a Key Pair

Please follow the intruction below to generate a key pair:

- Right-click the entry `Private Keys` in the navigation tree and select `Create Private Key` from drop-down menu or press the `Create Private Key` symbol in the function bar to open the wizard for creating a key pair. The following window opens and you can insert the mandatory parameters. Subsequent click **Finish** to create the key pair or **Next** to insert the subject parameters.



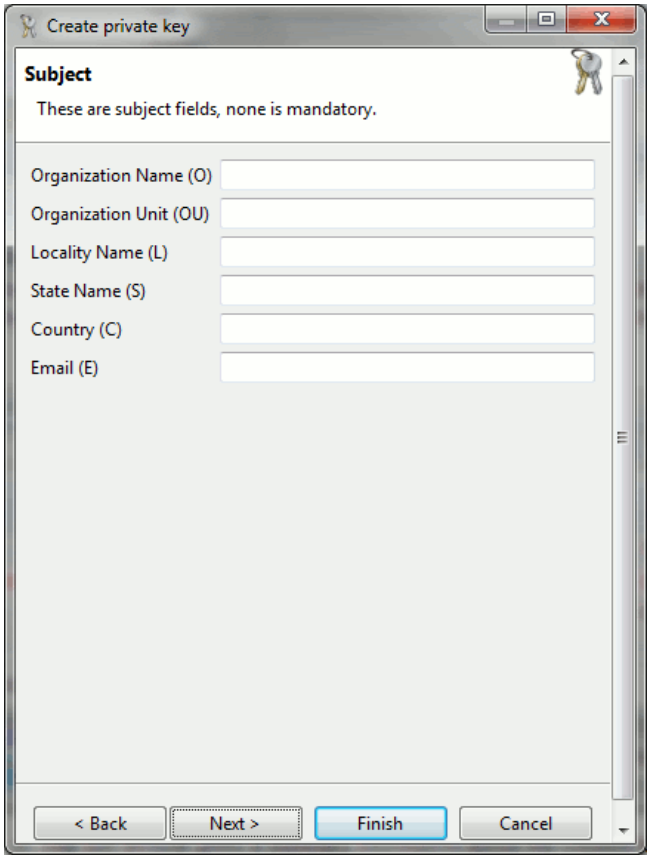
In table below you find a Description of the mandatory parameters:

Mandatory parameters

Parameter	Description
Stations ID	ID of the station the key pair is created for.
Label	Identifier for the functional assignment of the certificate.
Key Size	Specify the default value (1024 or a multiple of 1024).
Signature Algorithm	Hash Algorithm for digital signature. Possible-values: - SHA-1withRSA (default) - SHA-224withRSA - SHA-256withRSA - SHA-384withRSA - SHA-512withRSA Attention: Select a value higher than SHA-1 if the certificate is to be certified by Odette CA because SHA-1 is not accepted by Odette CA.
Valid from	Start of the keys validity date. Default: current date
Valid to	End of the keys valitity date. Dafault: current date plus tree years
Common Name (CN)	The Common Name parameter applies in case of a connection to an existing PKI (**P**ublic **K**ey **I**nfrastructure)

- The data of the next window are optional and concern your organization. Click **Finish** to create the key pair or **Next** to insert the extended

parameters.



- The extended parameters are optional and concerns the X.509 certificate. You need not change the present settings. (Further information can be found under the following URL: `http://tools.ietf.org/html/rfc5280`). Click **Finish** to create the key pair.
- The last window gives a summary information about the created key pair.

8.4.5 Use of CA Certificates

This chapter describes the procedure for working with CA (**C**ertificate **A**uthority) certificates. CA Certificates are issued by a TrustCenter like the Odette certification authority.

Generate CSR

- Generate a key pair as described in chapter 8.4.4 "Generate a Key Pair"
- Generate a certification request (CSR - **C**ertificate **S**igning **R**equest) via a right-click on the key pair in the list view and select `Create CA Request` or via the button **Create CA Request** at the bottom of the detailed view of the key. Define name and directory in which the request should be saved. By default the file is generated in PKCS#10 format.

- Send the request to the TrustCenter (CA). With Odette CA you can make your application online: `https://www.odetteca.com`. Please see `https://forum.odette.org/repository/odette-ca-help.pdf` for more information.

You receive your server certificate (your public key signed by CA), a CA root- and an intermediate certificate if all entries of your certificate order are correct.

Import of own certificates

- Import the Root certificate via a right-click on entry `CA Certificates` in the navigation tree and then select `Import CA Certificate` from the drop-down menu.
- Import the intermediate CA certificate a via right-click on `Intermediate Certificates` entry and single-click on `Import Intermediate Certificate` entry from the drop-down menu.
- Import the server certificate with the `Import CA Reply` entry which is reachable via a right-click on the private key in the list view. You can also execute this function via the `Import CA Reply` button in the key's detailed view.

Send the certificates to your partner and import the certificates of your partner as described below:

Import of the partner's certificate

- Import the Root certificate of your partner via a right-click on entry `CA Certificates` in the navigation tree and selecting `Import CA Certificate` from the drop-down menu.
- Import the intermediate CA certificate of your partner via right-click on `Intermediate Certificates` entry and selecting `Import Intermediate Certificate` entry from the drop-down menu.
- Import the server certificate of your partner via right-click on `Partner Certificates` entry in the navigation tree and selecting `Import Partner Certificate` from the drop-down menu.

Hints:

With TLS encryption either Root or Intermediate certificate must be imported.

With TSL encryption CertPath validation takes place by default. Please see chapter 22 "PKI Connection" for more information.

8.4.6 Export of the Own Public Key

Right-click the key pair in the list view and select `Export Certificate` from the drop-down menu to export your public key into a file (X.509 certificate). A window opens and you can specify directory and file name. Finish the export via a single-click on the **Save** button.

8.4.7 Import and Export of a Partner Certificate

rvsEVO enables to import trusted certificates from partners into the key management and to export already imported certificates from the key management into a file.

Import

Right-click the entry `Partner Certificates` in the navigation tree and select `Import Partner Certificate` from the drop-down menu for importing a partner certificate. A dialog box opens and you can input the following values: station ID for which the certificate should be imported, the certificate to be imported and a label. Press the **Import** button for importing the certificate.

Hint: The first certificate of a partner station is saved as default certificate by standard. This setting can be configured in parameter group `Keys` of the partner station. (See chapter 5.3.3 "Parameter Groups of the Partner Station:" for more information)

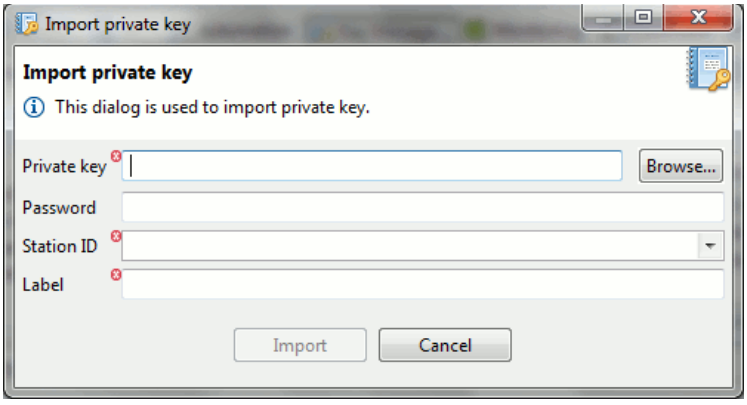
Export

Right-click the certificate in the list view and select `Export Certificate` from the drop-down menu for exporting a partner certificate into a file. A dialog box opens and you can input the name of the file in which the certificate should be saved. Finish the export via a single-click on the **Save** button.

8.4.8 Import and Export of a Private Key Pair

Import

- Right-click the entry `Private Keys` in the navigation tree and select `Import Private Key` from the drop-down menu for importing a private key or single-click the `Import Private Key` icon at the upper right border. The following dialog box opens:



- Select the key and the station ID for which the key should be imported. The private key can be imported for a local station, a virtual station or an internal partner station (if using OFTP File Service Module). Insert in input field **Password** the password of the key management file and in input field **Label** an identifier for functional assignment of the certificate usage. Press the **Import** button to import the key.

Hint: The first key pair is saved as default key as standard. This setting can be configured in parameter group `Keys` of the local station. (See chapter 5.2.2 "Parameter Groups of Local Station" for more information)

Export — Right-click the key to be exported in the list view and select `Export Private Key` from the drop-down menu. A dialog box opens and the directory and the name of the file is requested. Press the **Export** button to export the key into the specified directory and file.

8.4.9 Import and Export of ComSecure Public Keys

rvsEVO supports an appropriate format for the encryption, so it is necessary to convert a ComSecure public key into/from a X.509 certificate before importing/exporting it into the key management. X.509 is a standard format for digital certificates; certificate files end normally with .cer.

This functionalities can be executed via the functions `Import ComSecure Public Key` and `Export Certificate to ComSecure`.

Import — Right-click the `Partner Certificates` entry in the navigation tree and select `Import ComSecure Public Key` for importing a ComSecure public key as X.509 certificate into the rvsEVO key management.

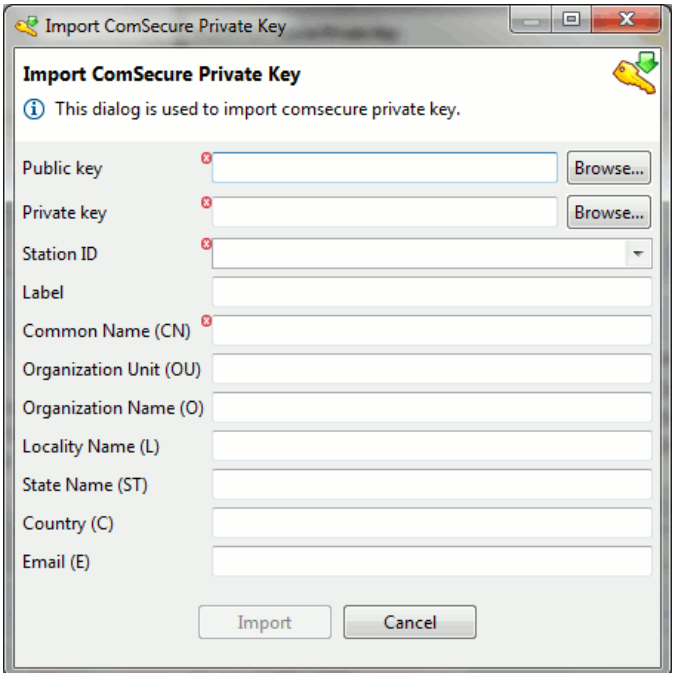
Export — Right-click a certificate in the list view and select `Export Certificate to ComSecure` to export a public key in X.509 format into a ComSecure format.

8.4.10 Import of a ComSecure Key Pair

This functionality is important for migration from rvs® portable to rvsEVO.

Right-click the `Private Keys` entry in the navigation tree and select `Import ComSecure Private Key` from the drop-down menu to open the dialog box shown below for importing a ComSecure key pair or press the icon `Import ComSecure Private Key` at the upper right border.

The private and public key and the station ID for which the key pair is imported can be selected. Further, you can insert a label for the functional assignment and some additional information.



8.4.11 Deletion of a Certificate / Private Key Pair

Right-click the certificate or key pair which should be deleted in the list view and select `Delete Certificate` / `Delete Key Pair` from the drop-down menu. Confirm the subsequent security query to remove the key.

8.4.12 Automatic Certificate Exchange via rvsEVO-GUI

The functionalities `ACX Deliver Certificate`, `ACX Request Certificate` and `ACX Replace Certificate` can be executed via the corresponding icons in the function bar.

Instruction for entering the parameter values:

Station IDs can be written into the input feld or selected from the combobox. If you enter the value dircetly into the input field you can use the wild-cards '*' and '?'.

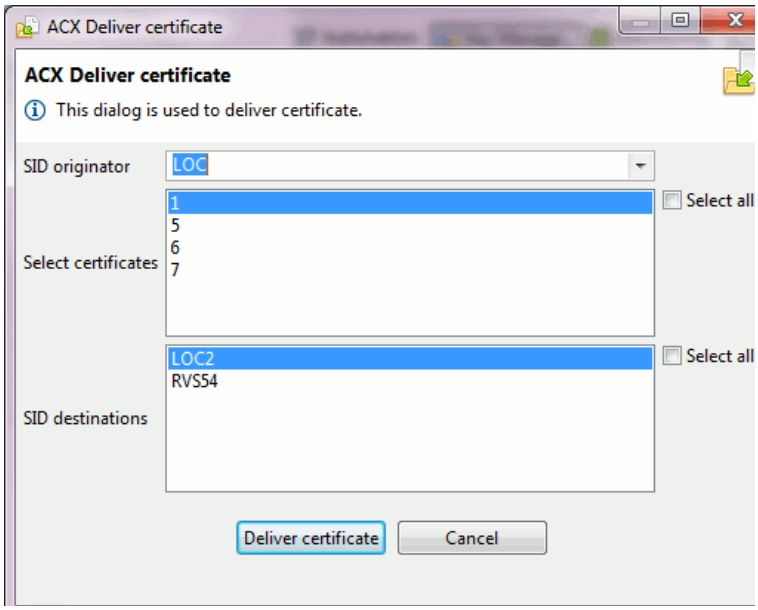
A checkbox is to be enabled or disabled by single-click in the checkbox.

Deliver Certificates

Use this feature to deliver a certificate to a partner station and import it at the partner station.

- Open the wizard for ACX Deliver certificate via a single-click on the following icon in the toolbar: 

.



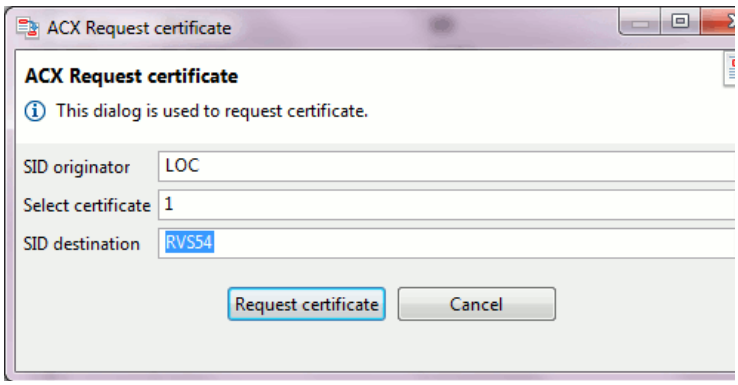
- Select the station ID of the certificate's originator via selection box `SID Originator`.
- Select the certificates to be delivered via `Select certificates` selection box. If the check box `Select all` is activated all certificates are delivered.
  Hint: Selection of the certificate is done via certificate index. You find the index in the list view and detailed view of a certificate.
- Insert the station IDs of the neighbours to whom the certificate is to be sent via input box `SID destinations`. If the check box `Select all` is activated the specified certificates are delivered to all neighbours.
- Right-click the **Deliver certificate** button to execute the process.

Request certificate

Use this feature to request a cerificate from the neighbour station and to deliver an own one with default label to this station.

- Open the wizard for `ACX Request certificate` via a single-click on the following icon in the tool bar  .

.



- Select the station ID of the certificate's originator via selection box `SID Originator`.
- Select the certificate to be sent via `Select certificates` selection box.
  Hint: Selection of the certificate is done via certificate index. You find the index in the list view and detailed view of a certificate.
- Select the station ID of the neighbour to whom the request is to be sent via selection box `SID destination`.
- Right-click the **Request certificate** button to execute the process.

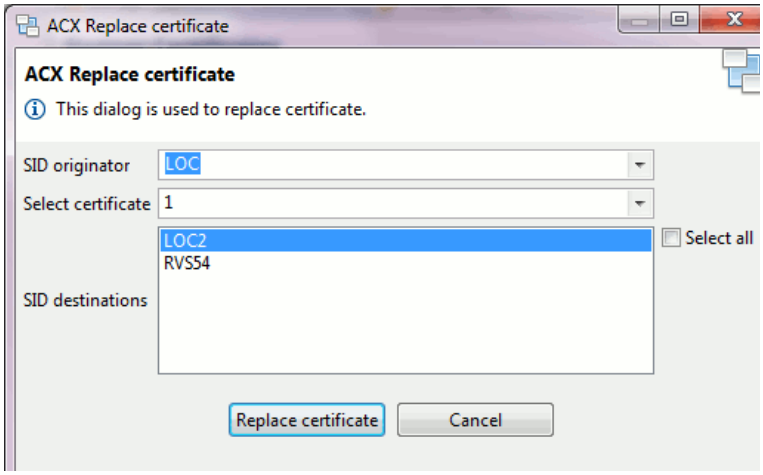
Hint: The imported certificate gets the default label by default. If there is already a certificate with default label, that one with the latest expiration date is used. If an other certificate is assigned the default label the settings can be changed in Stations perspective via parameter group Keys of local station / neighbour station.

Replace Certificate

Use this feature to send a certificate to a partner station and replace an already existing certificate in this station.

Open the wizard for `ACX Replace certificate` via a single-click on the following icon in the tool bar  .

.



- Select the station ID of the certificate's originator via selection box `SID Originator`.
- Select the certificate to be replaced via `Select certificate` selection box.
  Hint: Selection of the certificate is done via certificate index. You find the index in the list view and detailed view of a certificate.
- Insert the station ID of the neighbours whose certificate is to be replaced via input box `SID destination`. If the check box `Select all` is activated the specified certificate is replaced at all neighbours.
- Right-click the **Replace certificate** button to execute the process.

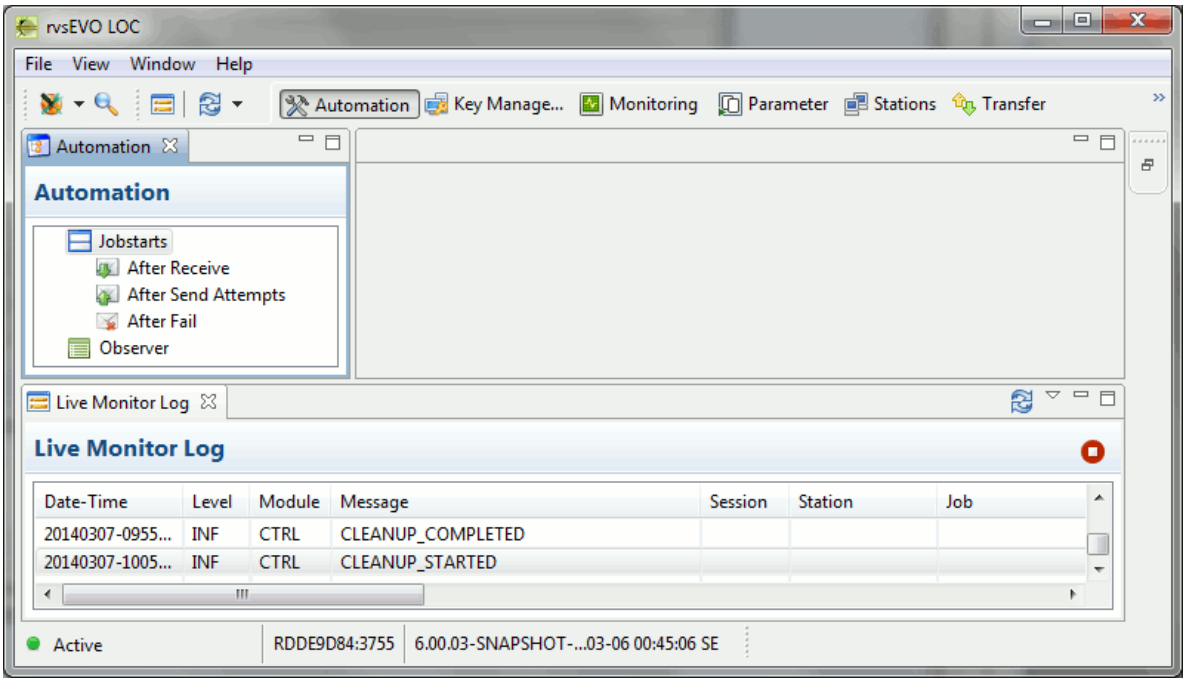
9 Automation with rvsEVO

In this chapter the jobstarts and the `Observer` (automatic generation of send jobs) are described.

9.1 The User Interface of Automation Perspective

Select the menu item View in the function bar and thereafter the subentry `Automation` to open the Automation perspective.

The illustration below shows the default view of Standard Edition.



The symbol in the function bar provides the following functionalities:

- Create Jobstart (page 120)
  - Jobstart After Receive
  - Jobstart After Send Attempts
  - Jobstart After Failure
- Create Observer (page 136)

Below, the navigation tree provides the following entries for list output:

- Jobstarts (all jobstarts)
  - After Receive
  - After Send
  - Jobstart after Failure
- Observer (automatic generation of send jobs).

Via a mouse-click on the corresponding entry in the navigation tree a list of jobstarts or Observer is output.

At the bottom of the GUI the `Live Monitor-Log` view is positioned.

9.2 Jobstarts

The Jobstart configuration comprises rules that allow special processes to be started when appropriate files are being sent or received or when a job fails.

Hint: We recommend to save your own scripts in `$RVS_HOME/jobs` directory which has no other use and is taken into consideration in case of backup and recovery. Don't save changed scripts and your own one in `$RVS_HOME/bin` directory or subdirectories of `$RVS_HOME/bin` because these will be overwritten during update installations.

Job filter

If more than one jobFilter applies to the send job, receive job or failed job, the job is started whose jobFilter applies most exactly (eg while using wildcards: `testdoc*` is more exact than `testd*`).

9.2.1 Creating a Jobstart

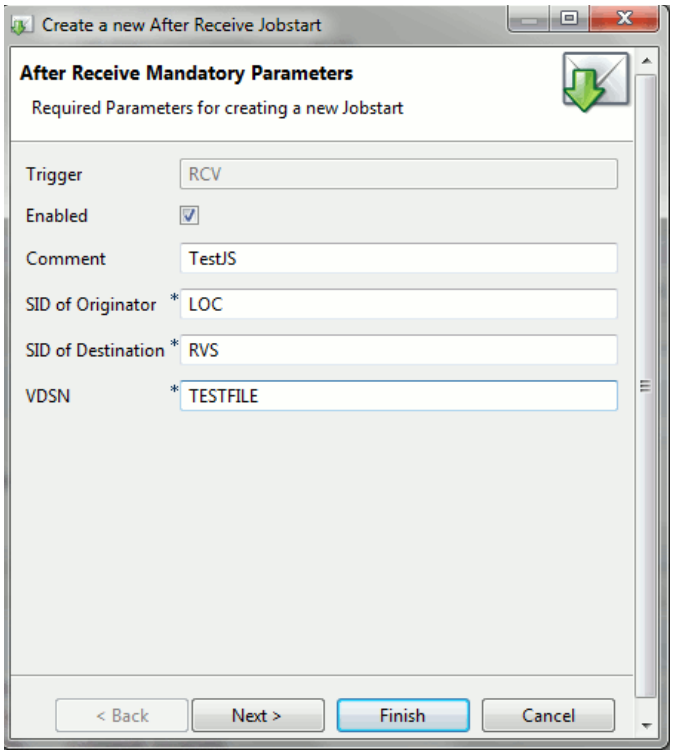
- A new Jobstart will be created via the symbol in the functionbar or with a right-click on the subentry `Jobstart After Receive`, `Jobstart After Send Attempt` or `Jobstart After Fail` in the Administration tree and then subsequently select `Add New Entry`.

Jobstart After Receive

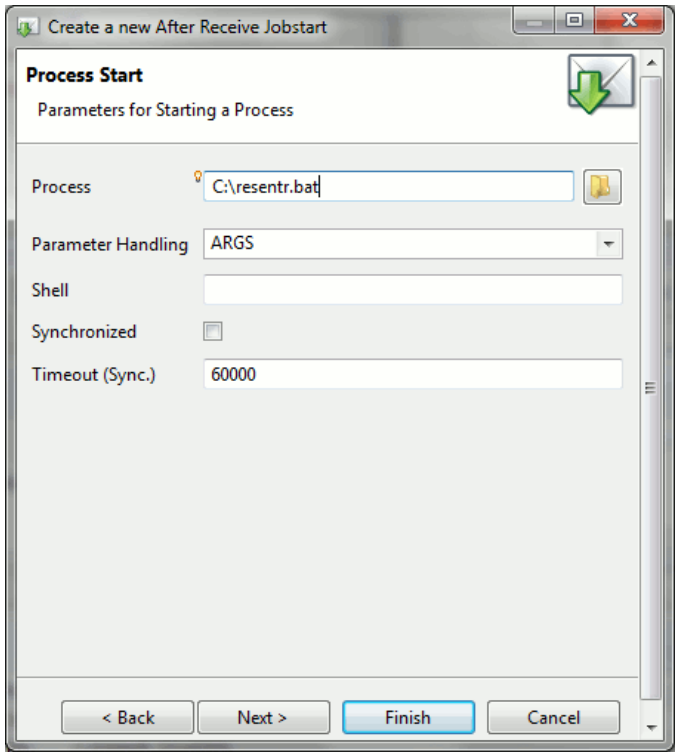
A Jobstart After Receive is triggered after file receive and before sending an EERP.

In the following example the program `C:/resentr.bat` will be started after the local station `LOC` receives a file with the virtual file name `TESTFILE` from the neighbour station `RVS`. The file is renamed to `TESTFILE_NEW` and saved in `$RVS_HOME/files/inbox` directory.

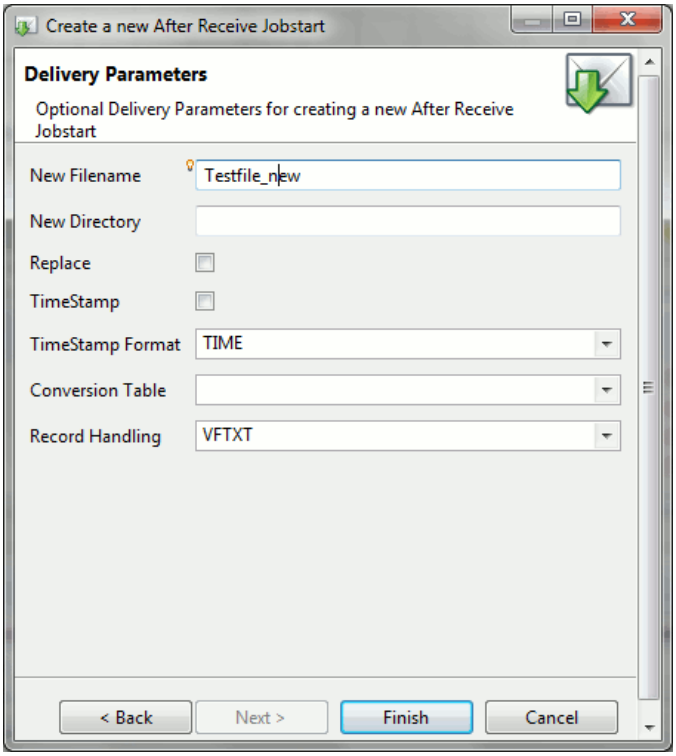
- In the first dialog the mandatory parameters must be entered (Enabled, Comment, SID of Originator, SID of Destination and VDSN).



- Press the **Finish** button to create the jobstart or **NEXT** to setup the parameters for starting a process (Process, Parameter Handling, Shell, Synchronized and Timeout (Sync.)).



- Press the **Finish** button to create the jobstart or **NEXT** to setup the delivery parameters (New Filename, New Directory, Replace, Time-Stamp, TimeStamp Format, Conversion Table and Record Handling).



- Press the **Finish** button to create the new Jobstart After Receive.

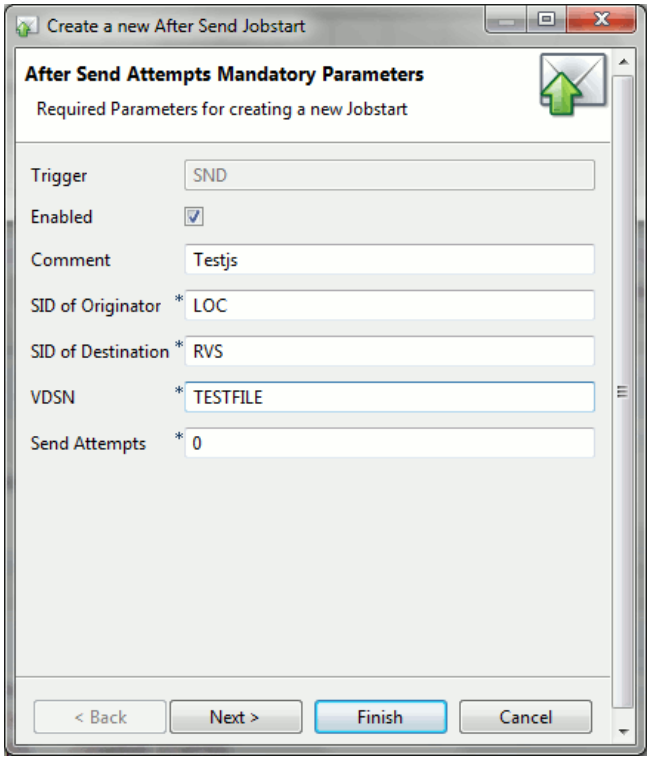
The last window displays the most important parameters of the jobstart.

Hint: Please see table “Jobfilter elements” on page 127 for the description of the parameters.

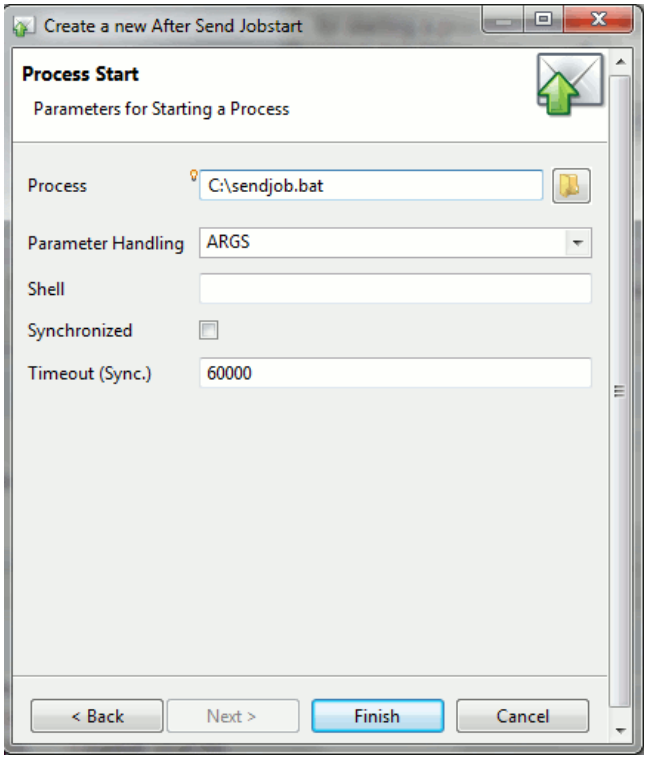
Jobstart After Send Attempts

In the following example the program `C:/sendjob.bat` will be started after a file with the virtual filename `TESTFILE` was successfully delivered (**Send Attempts=0**) from local station `LOC` to the neighbour station RVS.

- In the first dialog the mandatory parameters must be entered (Enabled, Comment, SID of Originator, SID of Destination,VDSN and Send Attempts).



- Press the **Finish** button to create the jobstart or **NEXT** to setup the parameters for starting a process (Process, Parameter Handling, Shell, Synchronized and Timeout (Sync.)).



- Press the **Finish** button to create the new Jobstart After Send Attempts.

The last window displays the most important parameters of the jobstart.

Hint: Please see table “Jobfilter elements” on page 127 for the description of the parameters.

Jobstart after Aborted Send Attempts

It is possible to define a number of aborted send attempts after those a job is to be started. Therefor set the parameter **Send Attempts**. If several jobs are to be started after a different number of send attempts this is handled as follows.

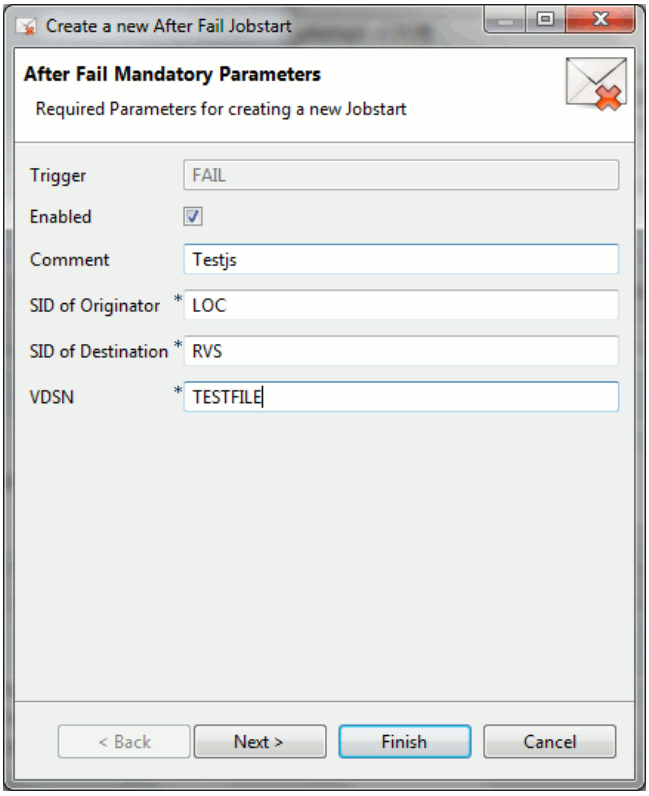
Example: Program 1 (P1) shall be started after one send attempt and program 2 (P2) shall be started after three send attempts, then:

1. send attempt -> P1
2. send attempt -> P1
3. send attempt -> P2
4. send attempt -> P1
5. send attempt -> P1
6. send attempt -> P2

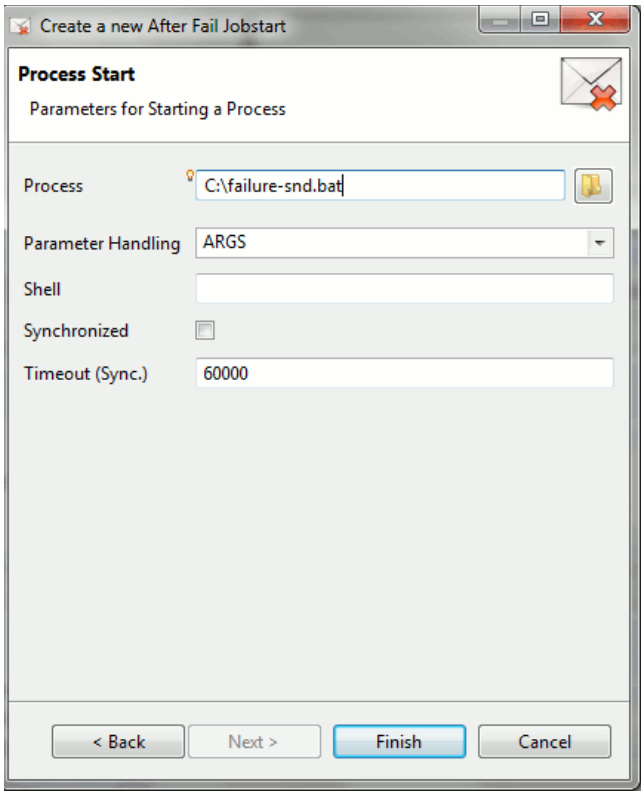
Jobstart After Fail

In the following example a file with the virtual filename `TESTFILE` should be send from local station `LOC` to neighbour station `RVS`. The program `C:\Failure-snd.bat` is to be started after receiving an error message after send attempt.

- In the first dialog the mandatory parameters must be entered (Enabled, Comment, SID of Originator, SID of Destination and VDSN).



- Press the **Finish** button to create the jobstart or **NEXT** to setup the parameters for starting a process (Process, Parameter Handling, Shell, Synchronized and Timeout (Sync.)).



- Press the **Finish** button to create the new Jobstart After Fail.

The last window displays the most important parameters of the jobstart.

Hint: Please see table “Jobfilter elements” on page 127 for the description of the parameters.

Note: A JobStart after fail is launched if a job was changed to `FAILED` status or if active jobs were changed into an error status which requires an operator intervention (e.g. `Restart`). A JobStart After Fail is started also after a NERP was received.

The following situations activate a Jobstart After Fail:

Send transmission:

– preprocessing fails (status: `SP_FAILED`)

– partner station is missing

– Jobstart After Send Attempts fails (`ENDED_WITH_JS_ERROR`)

– SFNA (**S**tart **F**ile **N**egative **A**nswer) after connection establishment (e.g. unencrypted file transmission while the neighbour station demands encryption)

– `FATAL_ERROR` after SFNA (e.g. the file does not exist)

– Jobstart fails after successful send attempt (`ENDED_WITH_JS_ERROR`)

– EFNA error (**E**nd **F**ile **N**egative **A**nswer) after transmission

– receipt of NERP (Negative-End-to-End-Response)

Receive transmission:

– dispatch of SFNA

– dispatch of EFNA (e.g. the byte count of the received file is not conform to the byte count of the transmitted file)

– subsequent treatment fails; e.g. missing private key for decryption (`SP_FAILED`)

– JobStart after receive fails (`ENDED_WITH_JS_ERROR`)

9.2.2 Jobstart Parameters

Jobfilter elements

Parameter	Description
Comment	free Text
Conversion table	For ASCII(ANSI) - EBCDIC conversion the following conversion tables are available: `ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273`. For EBCDIC - ASCII(ANSI) conversion the following conversion tables are available: `IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI` For using your own conversion table, please read the note for conversion tables on page “Conversion tables” on page 133
Direction	Defines filter rules for the communication direction. Possible values: - `SND` (when sending files) - `RCV` (when receiving files) - `FAIL` (after processing error).
Enabled	With this parameter you decide whether the job is to be started or not.
New Directory	With this parameter you can save the received file in another directory than `Inbox` directory.
New Filename	With this parameter you can save the received file with another name. VDSN is used by deault. Timestamp is added like defined in **TimeStamp** parameter.

Jobfilter elements

Parameter	Description
Parameter Handling	With this parameter you can decide how to transmit the job data to the process. Possible values: ARGS: job data are passed as arguments (default) ENV: job data are set as environment variables REPLACE: for compatibility with rvs® portable. For further information see chapter “Parameter Handling” on page 130
Process	Program to be started when all filter conditions apply. Parameters, seperated by spaces, can be passed to the program. The following must be noted: - Terms with blanc should be placed in quotation marks. - By using cmd /: indicate `cmd.exe` only with the script name. - If **Parameter Handling** is ARGS: indicate the user parameters first and thereafter the system parameters. - With Linux shells the parameters are passed without quotation marks, with Windows the parameters are passed with quoations marks. A defined set of the following job parameters is passed to the programs: - 1 JobId - 2 Station ID (of sender or recipient) - 3 File name of the file sent or received - 4 VDSN - 5 Date of the job - 6 Time of the job - 7 Number of attempts to send. - 8 Format of a file - 9 Record length of a file - 10 Transmitted Bytes For explanation of send parameters see chapter 6.2.1 "Send Parameters". How these parameters should be used in a script, see the example at the end of this chapter (file `$RVS_HOME/bin/jobstart/jobstart_detailed.bat`). (not for Remote GUI; please see parameter **process (Server)**).
Process (Server)	For Remote GUI (instead of parameter **Process**)

Jobfilter elements

Parameter	Description
Record Handling	With this parameter you decide, whether each line of a received file in **F** or **V** format is terminated by line feed. Possible values: VFTXT: Files in F or V format are saved in text mode. Line feed is inserted at the end of each record (default) VTXT_FBIN: Files in V format are saved in text mode; files in F format are saved in binary mode.
Replace	With parameter **Replace** you decide how to handle receiving files which own the same name like existing files. If the checkbox is activated the existing file is replaced. Otherwise a timestamp is added (default).
Send Attempts	Number of failed attempts to send. Successful file transmission is indicated by "0" here.
Shell	Command Shell for executions of the program e.g. ksh,.csh, ... on Unix systems
SID of Destination	Station Id of target station
SID of Originator	Station Id of source station (local or virtual Station)
Synchronized	With this parameter you decide whether the OFTP session between sender station and receiver station will be kept alive waiting until the process is finished or not. If the checkbox is activated the session will be kept alive. Default: not activated.
Timeout (Sync.)	If parameter **Synchronized** is activated: Time Out in milliseconds after which the communication program the connection closes, if the partner station does not answer. Default: 60 000
TimeStamp	With this parameter you define rules for timestamp creation. If the parameter is ... activated: timestamp is to be generated generally ... not activated: timestamp is to be added only if necessary (default)

Jobfilter elements

Parameter	Description
TimeStamp Format	Format of the timestamp. Possible values: - `TIME`: creation time of the job and counter; format: `hhmmssccc` (default) - `DATETIME`: creation date and time of the job and counter; format: `YYMMDDhhmmssccc` - `SFID_DATETIME`: Odette timestamp of SFID; date, time and counter in format `YYMMDDhhmmsscccc` - `COUNTER`: 000000 - 999999. If the counter is not sequential, the gap is filled first before the counter is counted up.
VDSN	Virtual file name (regular expression as filter)

Parameter Handling

Hint: Subsequently you can find a detailed description of the alignment **Parameter Handling**=`ENV` and **Parameter Handling**=`REPLACE`.

Parameter Handling=ENV

Parameter Handling=`ENV`: job data are set as environment variables. The following environment variables are specified (in angle brackets: name in XML file, in brackets: GUI name):

- **RVS_COMPRESSION** <compression> (Compression)
- **RVS_CONVERSION_TABLE** <conversionTable> (Code Table)
- **RVS_CREATED_AT** <creationDate> (SFID Date/Time)
- **RVS_DATE** Date of <creationDate>
- **RVS_DESCRIPTION** <fileDescription> (File Description)
- **RVS_DIRECTION** <direction> (Direction)
- **RVS_DISPOSITION** <disposition> (Disposition)
- **RVS_ENCRYPTION** <encryption> (Encryption)
- **RVS_ENCRYPTION_ALGORITHM** <encryptionAlgorithm> Encryption Algorithm)
- **RVS_ERROR_ID** <errorID> (Error Number)
- **RVS_ERROR_TEXT** <errorText> (Description)
- **RVS_EXTERNAL_JOB_ID** <externalJobId> (External Job ID)
- **RVS_FILE_NAME** <filename> (Filename)
- **RVS_FILE_NAME_SRC** <filenameSrc> (Source File Name)
- **RVS_JOB_ID** <jobNumber> (ID)
- **RVS_LABEL** <label> (Label)
- **RVS_LAST_CHANGE** <lastStateChange> (Last Change)

- **RVS_NERP_CREATOR_ODETTE_ID** <nerpCreatorOdetteID> (Nerp Creator OdetteID)
- **RVS_NERP_REASON_CODE** <nerpReasonCode> (Nerp Reason Code)
- **RVS_NERP_REASON_TEXT** <nerpReasonText> (Nerp Reason Text)
- **RVS_RECORD_FORMAT** <recordFormat> (Format)
- **RVS_RECORD_LENGTH** <recordLength> (Record Length)
- **RVS_RESTART_POSITION** <restartPos> (Restart Position)
- **RVS_SECURITY_FEATURE_SET** <securityFeatureSet> (Security Feature Set)
- **RVS_SEND_ATTEMPTS** <sendAttempts> (Send Attempts)
- **RVS_SERIALIZE** <serialization> (Serialization)
- **RVS_SID** <SID>
- **RVS_SID_DESTINATION** <sidDestination> (SID Destination)
- **RVS_SID_ORIGINATOR** <sidOriginator> (SID Originatior)
- **RVS_SIGNATURE** <sign> (File Signature)
- **RVS_SIGNATURE_EERP** <signEERP> (Request Signed EERP/NERP)
- **RVS_STATUS** <status> (Status)
- **RVS_TIME** Time of <creationDate>
- **RVS_TIME_START_FILE** <timeStartFile> (Time Start File)
- **RVS_TRANSMITTED_BYTES** <transmittedBytes> (Bytes Transmitted)
- **RVS_VDSN** <VDSN> (VDSN)

The table “Explanation of the job attributes” on page 182 shows the description of the environment variables.

Parameter Handling=REPLACE

Parameter Handling=REPLACE: A copy of the denoted file is created and discarded into `temp` directory. The placeholders of the scripts are substituted for job data. The placeholders are encased by ? (e.g. `"move ?DSN? /home/rvsevo/incomming" becomes "move /home/rvsevo/rvsEVO/files/inbox/TEST.TXT /home/rvsevo/incomming"`).

With MAINTENANCE parameter **PermissionTempScript** you can set up the file permissions of the temporary file. (Default: read/write/execute-permission for owner and group).

Placeholder of all jobstarts:

- **?DSN?:** DataSetName (name of local data set, where received information has been stored)

- **?VDSN?:** Virtual data set name under which the data set was transmitted
- **?DTAVAIL?:** Date, when the data set was available for sending; Format `TT/MM/JJ HH:MM:SS`
- **?FORMAT?:** Record format of data set
- **?BYTES?:** Number of transmitted bytes
- **?RECORDS?:** Number of transmitted records for F and V format data sets; always zero for **T** und **U** format data sets.
- **?DTRCV?:** Date, when data set was delivered to local user in format `TT.MM.JJ HH:MM:SS`
- **?UID?:** User ID
- **?SID?:** Station ID
- **?DSNTEMP?:** Name of temp. data set

Placeholder of jobstart type SND:

- **?MAXRECL?:** The meaning of this field depends upon the record format of the transmitted data set:
- - F format: length of each record
- - V format: maximum length a record may have
- - T und U format: always 0 (zero)
- **?LABEL?**: string if the send command contained a LABEL parameter. Can be used to identify the send command.
- **?SECN?**: rvsEVO JobID (number of send command SE in rvs® portable)
- **?SKCN?**: rvsEVO JobID (number of send command SK in rvs® portable)
- **?SIDORIG?**: StationID of originatior
- **?SENDATT?**: number of unsuccessful attempts after which the program is to be started

Placeholder of jobstart type RCV:

- **?MAXRECL?**: The meaning of this field depends upon the record format of the received data set:
- - F format: length of each record
- - V format: maximum length a record may have
- - T und U format: always 0 (zero)
- **?CNQS?:** rvsEVO JobID (command number of EERP (End-to- End-Response) for received file in rvs® portable).
- **?CNIE?:** rvsEVO JobID (command number of IE for received file in rvs® portable).

– **?CNIZ?:** rvsEVO JobID (command number of IZ for received file in rvs® portable)

Conversion tables

Note: The next section is a short explanation of the conversion tables, that are offered by rvsEVO.

Text files are stored on most systems in one of two computer codes, namely ASCII (American National Standard Code for Information Interchange) or EBCDIC (Extended Binary Coded Decimal Interchange Code).

ASCII is the standard code for UNIX and DOS/Windows systems. EBCDIC was developed for IBM Mainframe computers.

ASCII: US-ASCII ISO 646; the ASCII character set defines 128 characters (0 to 127 decimal). This character set is a subset of many other character sets with 256 characters, including the ANSI character set of MS Windows.

ANSI: Windows ANSI, Values 0 to 127 are the same as in the ASCII character set, values 128 to 255 are similar to the ISO Latin-1 character set.

EBCDIC 037: support characters, which are used in the following countries: Australia, Brazil, Canada, New Zealand, Portugal, South Africa, USA.

EBCDIC 273: supports characters (especially umlauts), which are used in the following countries: Germany, Austria and Switzerland.

How to add your own conversion table:

In order to use your own conversion table, please follow the procedure given below:

- create a new conversion table. The content of this file is described in the following XML schema:

```
<conversiontable codein=Original-Characterset
codeout=Ziel-Characterset>
    <description>Table to converts signs from ... to
...</description>
     <char value="0">0</char>
     <char value="1">1</char>
            .
            .
</conversiontable>
```

  Note: `char value` is equivalent to the source code and the value in angle brackets is equivalent to the target code.

- Store the XML file in the directory `$RVS_HOME/conf/conversiontables`

- To add your table to the list of conversion tables, edit the XML file `$RVS_HOME/conf/conversiontables/characterSetConverters.xml`.

  Next you find an abridged version of the file

  `characterSetConverters.xml:`

```
<characterSetConverters coding="">
    <!-- -->
    <converter cid="ANSIIBM037"
        >conversion_ANSI_IBM037.xml</converter>
    <converter cid="IBM273-ASCII"
        >conversion_IBM273_ASCII.xml</converter>
        .
        .
</characterSetConverters>
```

  In the example above `cid="ANSI-IBM037" (cid="IBM237-ASCII")` is the used name in rvsEVO and `conversion_ANSI_IBM037.xml (conversion_IBM237_ASCII.xml)` is the name of your XML file.

- Please restart rvsEVO after your complement.

9.2.3 Displaying Jobstarts

Jobstart lists

Listings of jobstarts can be output via a single-click on the appropriate entry in the navigation tree. The list is displayed in a new view, by default at the bottom of the window. It can be opened several jobstart lists. The view in foreground is to be selected via a mouse-click on the tab. The following information is given: ID (Job ID awards by rvsEVO in format YYMMDDhhmmssccc), Direction, VDSN, SID Originator, SID Destination, Enabled and Send Attempt.

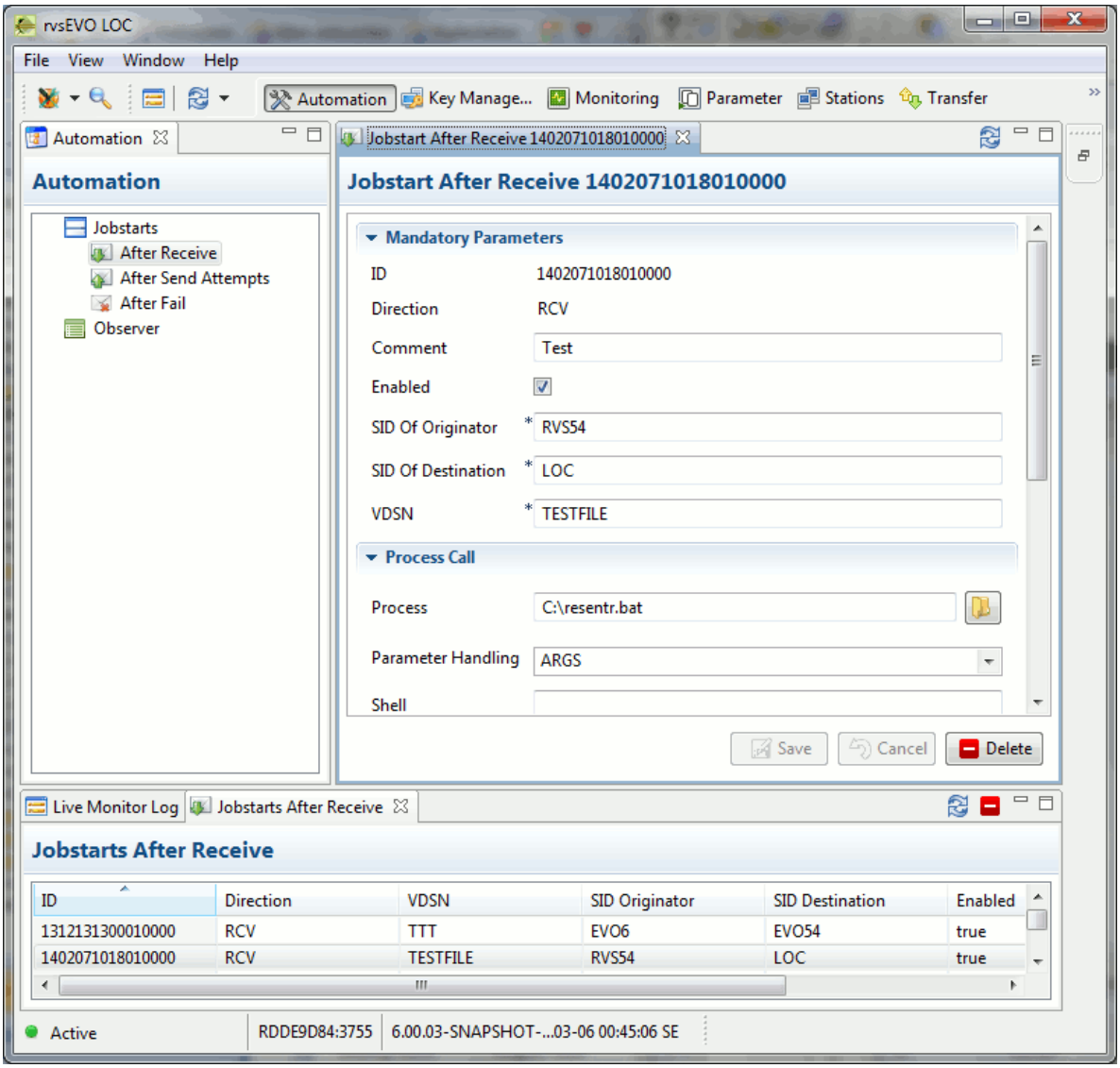
With a single-click on the column headings you get a sorted output. The list can be refreshed via an icon at the top right edge.

Via greyed out row beneath the column header you can set output filters. The wildcards '*' (for one or more signs) and '?' (for one sign) are permitted. Via 'Reset Filter' symbol you can switch off the filter function.

Detailed view

With a double-click on an entry in the jobstart list you get the detailed view of the jobstart. Another way to output the jobstart details is to right-click the jobstart and thereafter select `Open in New View`. In the second case the detailed view of several jobstarts can be opened . The view in foreground is to be selected via tab.

The illustration below shows the default view on the right-hand side and the list view at the bottom of the window.



9.2.4 Edit a Jobstart

The parameters of a jobstart can be edited in the detailed view. Save your changes via a mouse-click on the **Save** button. After changing the parameters you cannot open the detailed view of another jobstart without saving or canceling the changings.

9.2.5 Delete a Jobstart

There are two ways to delete a jobstart: Either by opening the detailed view of the jobstart and thereafter clicking on **Delete** button or by right-clicking the jobstart in the list and thereafter selecting **Delete Jobstart**, or pressing the delete icon at the top right edge. To delete several jobstart select the jobstarts to be deleted while keeping the **ctrl** button pressed and select the jobstarts to be deleted with a left mouse-click. Thereafter

you can delete the jobstarts via a right-click or via the icon at the top right edge.

9.3 Observer

The feature Observer generates send entries for files, which are in a configurated directory. The observer checks in regular time intervals a directory for the specified file names (mask). If such files are found, the send entry will be created. The options for send entry can be configurated, too. It is similar to the feature `rvsjs` in rvs® portable.

9.3.1 Create an Observer

Instruction for entering the parameter values:

If the input field includes a drop-down arrow you must select one of the specified values from the combobox.

In input fields without a drop-down arrow you must directly write the value into the input field.

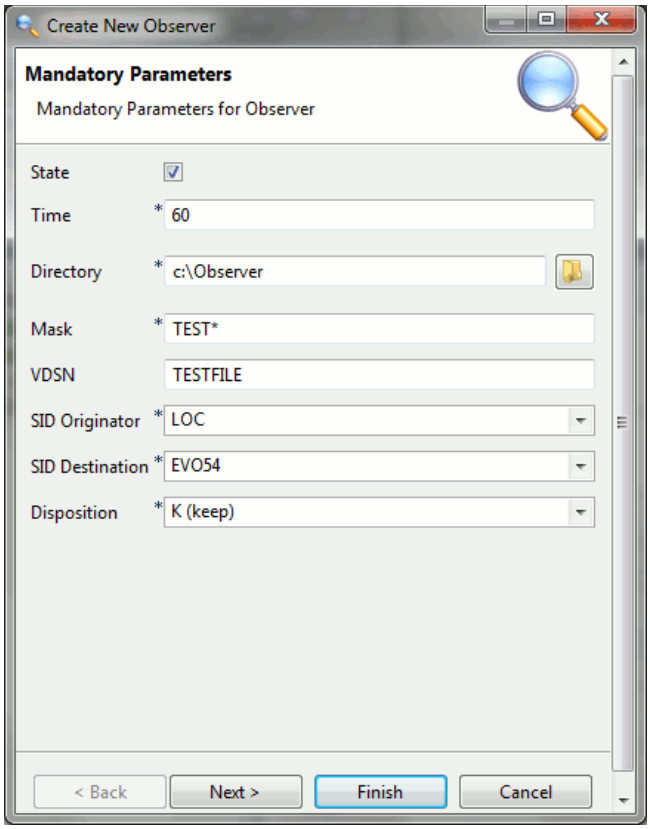
Exeption: Station IDs can be written into the input feld or selected from the combobox.

A checkbox is to be enabled or disabled by single-click in the checkbox.

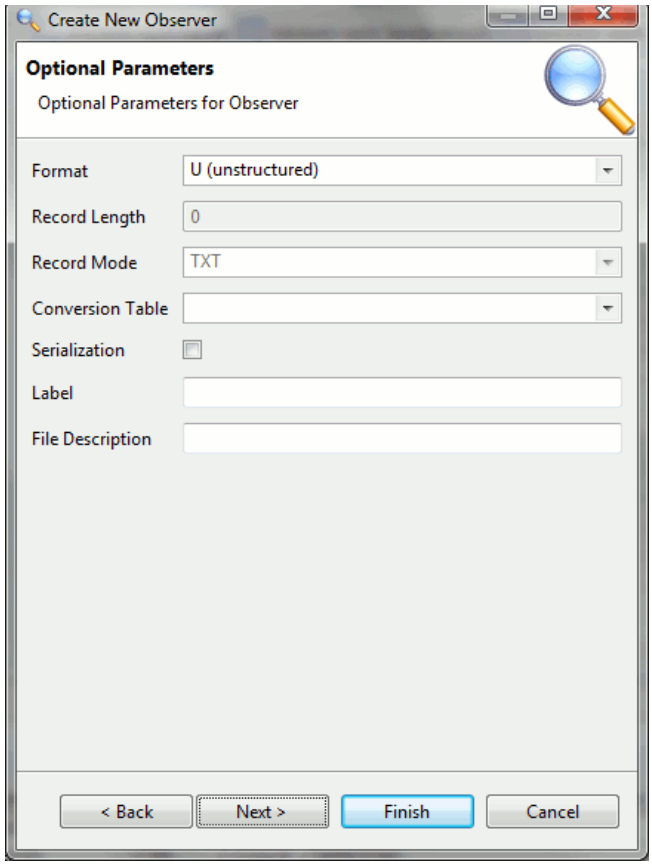
To create an Observer proceed as follows:

- Right-click the Observer entry in the navigation tree and select `Add Observer` from the drop-down menu or press the `Add Observer` symbol in the function bar to open the wizard for creating an Observer. The following window opens and the Mandatory Parameters (State, Time, Directory, Mask, VDSN, SID Originator, SID Destination, Disposition) are to be entered.

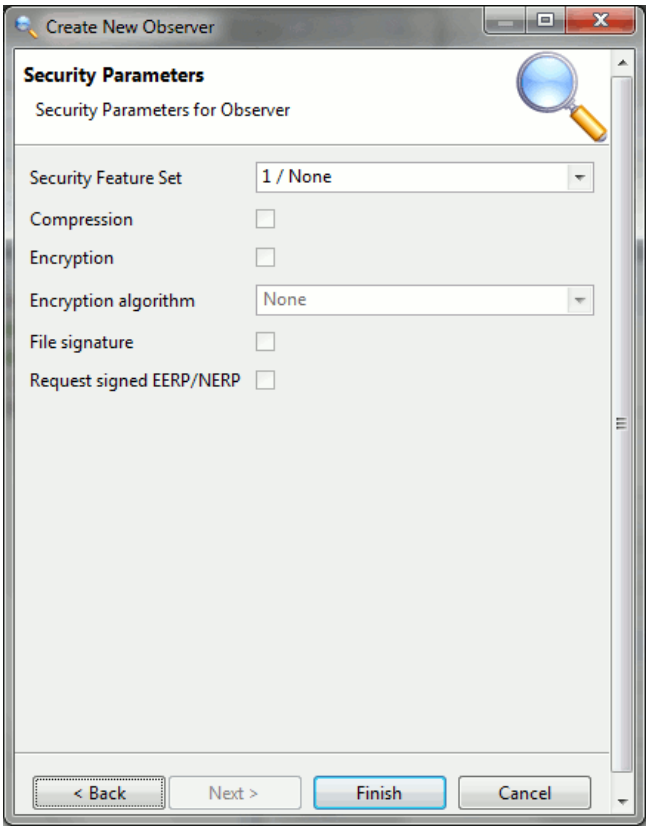
  In the following example the directory `c:\Observer` will be scanned every 60 seconds for all files transfered from neighbourstation EVO54, which file name begins with `TEST` and VDSN is `TESTFILE`.



- Create the Observer via mouse click on **Finish** button or press **Next** to insert the Optional parameters (Format, Record Length, Record Mode, Conversion Table, Serialization, Label and File Description).



- Create the Observer via mouse click on **Finish** button or press **Next** to insert the Security Parameters (Security Feature Set, Compression, Encryption, Encryption algorithm, File signature and Request signed EERP/NERP).



- Create the Observer with a single-click on **Finish** button. In the next dialog you are given a brief overview of selections you have made.

The last dialog informs about the successful creation of a new Observer.

In the following table the observer parameters are described.

Observer parameters

Parameter	Description
Enabled	This parameter defines, if the feature observer is active or not.
Time	Time interval between two checks.
Directory	Directory, which should be scanned.
Mask	The regular expression, which should be applied. The wildcards '?' (for one sign) and '*' (for one or more signs) and are permitted. If it finds them, it will create a send entry.

All other parameters from an Observer panel are for the sending a file.

Please see chapter 6.2.1 "Send Parameters" for further information.

Notes:
You can use the wildcard '*' in the Observer parameter **VDSN** for taking

over segments of the original filename, which are seperated by points, into the new VDSN.

Examples:

Original filename:	AAA.BBB.CCC
VDSN (Observer):	PART1.*.PART2
New VDSN:	PART1.BBB.PART2

Original filename:	DEH.KMUKQY.AS.111.ASN.D123.T456
VDSN (Observer):	DEH.KMUKQY.AS.*.ASN
New VDSN:	DEH.KMUKQY.AS.111.ASN

Original filename:	DEH.KMUKQY.AS.111.ASN.D123.T456
VDSN (Observer):	*.*.*.*.*
New VDSN:	DEH.KMUKQY.AS.111.ASN

You can also use the original filename as file description. This function is controlled by the properties file `$RVS_HOME/conf/rvs-system.properties` with the definition `observer.sfiddesc.usefilename=true`.

9.3.2 Displaying Observer

Observer lists

Observers can be listed via a single-click on the `Observer` entry in the navigation tree. The list is displayed in a new view. The following information are given: ID (Number awards by rvsEVO in format `YYMMDDhhmms-scccc`, State, Directory, Mask, VDSN, SID Originator, SID Destination).

With a single-click on the column headings you get a sorted output. The list can be refreshed via refresh icon at the top right edge.

Via greyed out row beneath the column header you can set output filters. The wildcards '*' (for one or more signs) and '?' (for one sign) are permitted. Via 'Reset Filter' symbol you can switch off the filter function.

Detailed view

With a double-click on an entry in the Observer list you get the detailed view of an Observer. Another way to output the jobstart details is to right-click the Observer and thereafter select `Open in New View`. In the second case the detailed view of several jobstarts can be opened . The view in foreground is to be selected via tab.

9.3.3 Edit an Observer

The parameters of an Observer can be edited in the detailed view. Save your changes via a mouse-click on the **Save** button. After changing the parameters you cannot open the detailed view of another jobstart without saving or canceling the changings.

9.3.4 Delete an Observer

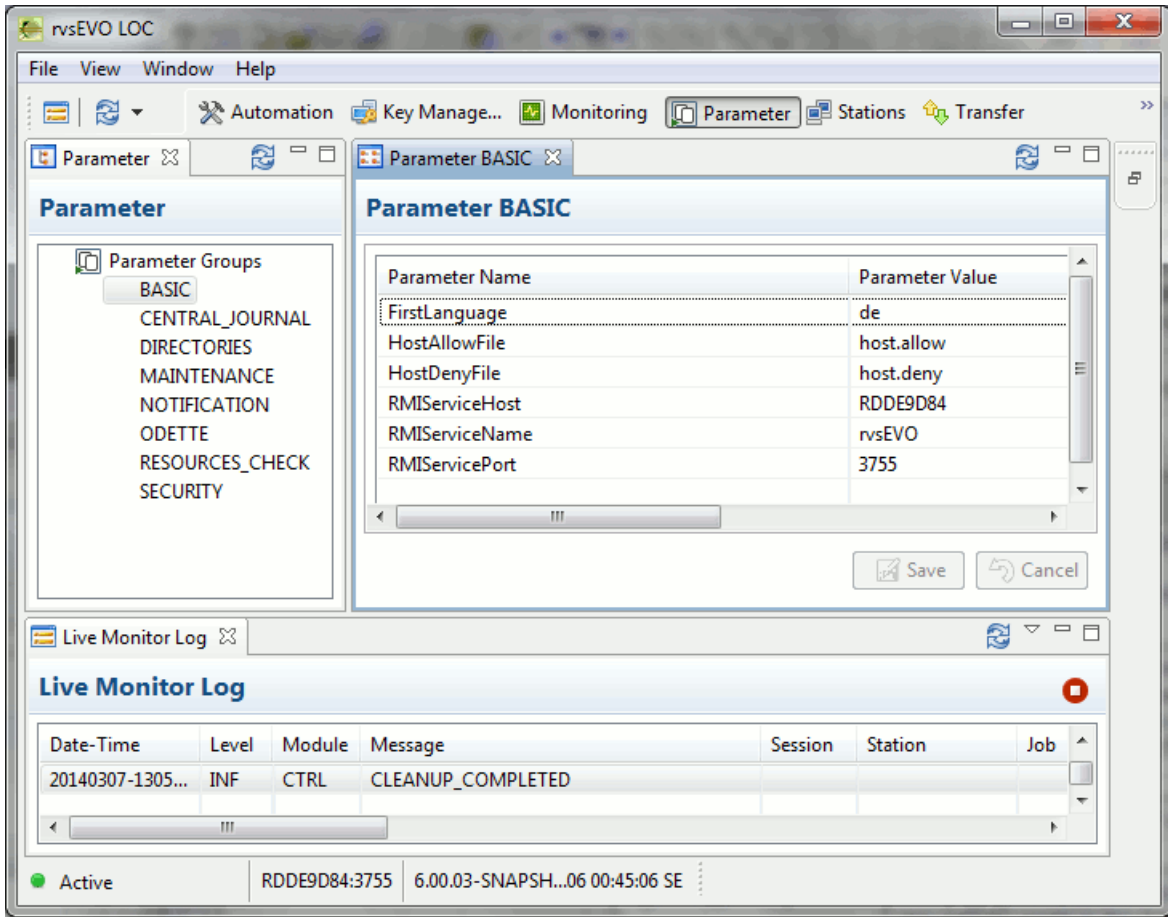
There are two ways to delete an Observer: Either by opening the detailed view of the jobstart and thereafter clicking on **Delete** button or by right-clicking the Observer in the list and thereafter selecting **Delete Observer**. or pressing the delete icon at the top right edge. To delete several Observers select the Observer to be deleted while keeping the **ctrl** button pressed and select the Observer to be deleted with a left mouse-click. Thereafter you can delete the Observer via a right-click or via the icon at the top right edge.

10 Customizing the Global rvsEVO Parameters

The present chapter describes how to customize rvsEVO.

10.1 The User Interface of Parameter Perspective

The parameter management is reachable via menu item `View -> Parameter`. The illustration below shows the default view of parameter perspective in Standard Edition:



The navigation tree on the left side of Parameter View contains the following entries:

- BASIC: general parameters
- CENTRAL_JOURNAL: parameters for `Central Journal`
- DIRECTORIES: paths of directories which includes the transfer and jobdata
- MAINTENANCE: parameters for archiving of completed or failed jobs.
- NOTIFICATION: parameters refering to the feature `rvs®-SNMP-Agent`

– ODETTE: OFTP Parameters
– RESOURCES_CHECK. Parameters for error handling relating to resource check and expiring certificates.
– SECURITY

Via a single-click on a parameter group the datailed view of this group is shown on the right side of the window. The parameters can be set up in this view.

10.2 BASIC parameters

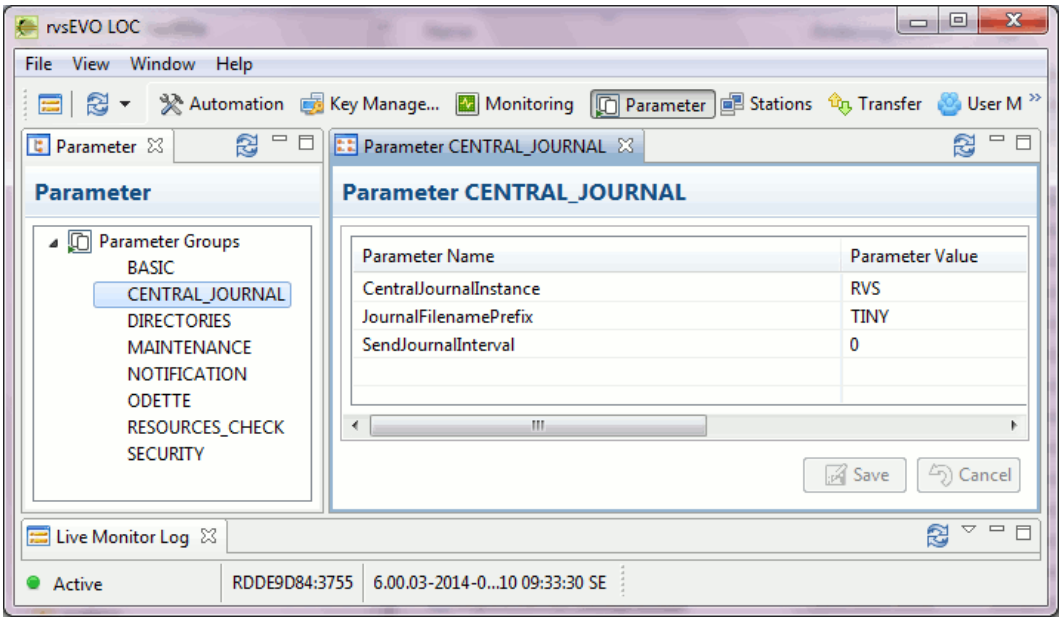
In this group the basic setting of rvsEVO can be carried out.

Parameter for Basic Settings

Parameter	Description
FirstLanguage	Language of the Server. Possible values: `de` (German), `en` (English).
HostAllowFile	Configuration file containing DNS names or IP addresses of hosts that may send rvsEVO commands to the rvsEVO server
HostDenyFile	Configuration file containing DNS names or IP addresses of hosts that may not send rvsEVO commands to the rvsEVO server.
RMIServiceHost	Host Name in the RMI registry. Default: `localhost`. RMI (Remote Method Invocation) is a protocol for the internal process communication in Java
RMIServiceName	Name of the rvsEVO service in the RMI registry. Default: rvsEVO
RMIServicePort	Port number for the communication in the RMI registry. Default: `3755`

10.3 CENTRAL_JOURNAL

This group contains the parameters for the feature `Central Journal`. The `Central Journal` functionality enables the central output of all data of transmitted files.



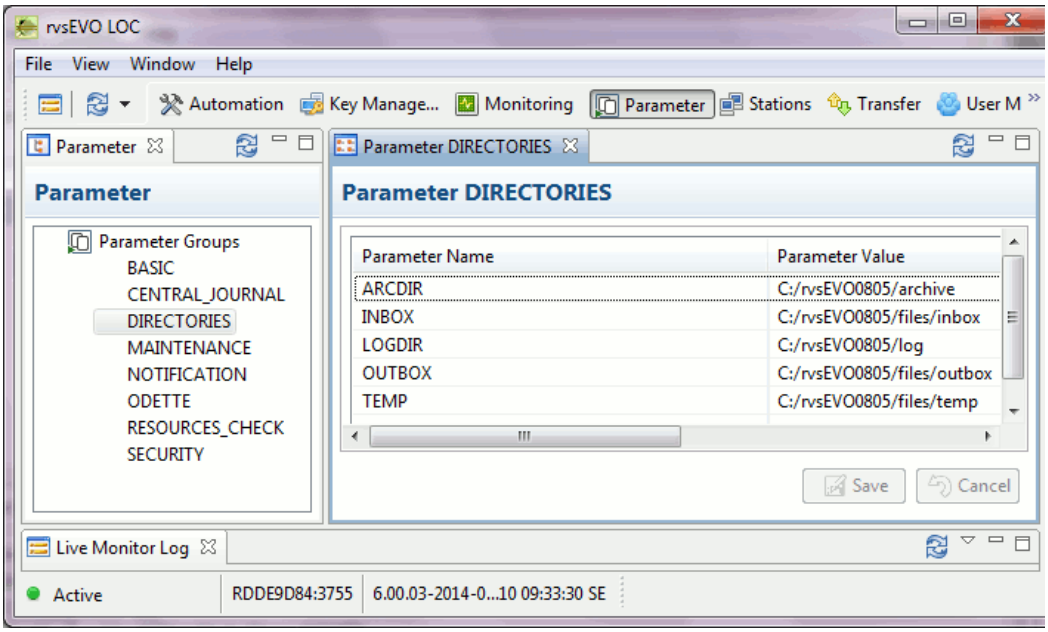
Please read the user manual „Central Journal“ for further information.

Parameters for Central Journal“

Parameter	Description
SendJournalInterval	Time interval in seconds between sending of two Journal files to the rvs® destination station (defined by the CentralJournalInstance parameter). No Journal file will be sent if no value or „0“ is specified.
JournalFilenamePrefix	The prefix for the Journal file name: default TINY.
CentralJournalInstance	The rvs® destination station that is to receive the Journal files. This station must be present in the rvsEVO station list.

10.4 DIRECTORIES

In this group you can indicate the directory paths for your data.

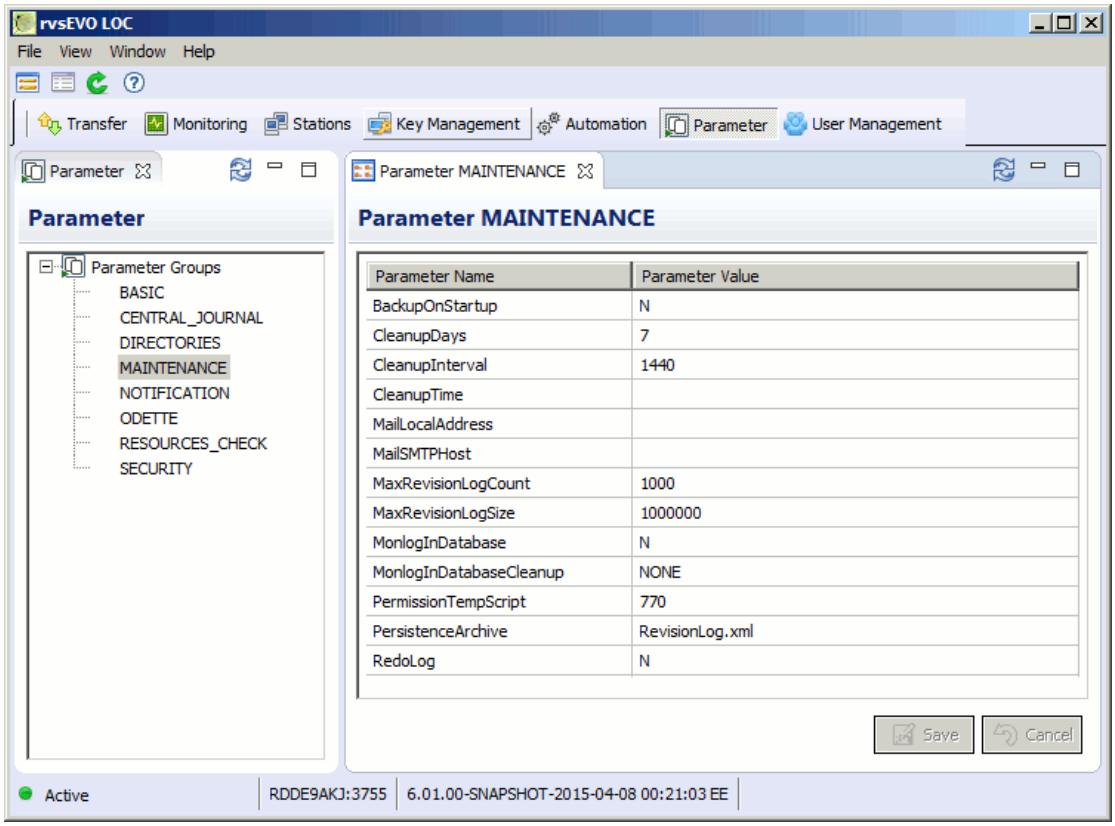


Parameters of Directories Group

Parameter	Description
OUTBOX	Temporary directory for files to be sent. Default: `$RVS_HOME\files\outbox`
TEMP	Directory for temporary use.
INBOX	Directory where completely received files are stored. Default: `$RVS_HOME\files\inbox`
LOGDIR	Directory for both log files: `startServer.log` and `rvsservice.log`.
ARCDIR	Archive directory where archive files and backup files are stored. Default: `$RVS_HOME\archive`. `RevisionLog.xml` files contain the entries of processed send or receive jobs (please see chapter 4.11). `.jar files` contain the backup data.

10.5 MAINTENANCE

In this group the parameters for security of important data and deletion of data from the database can be defined. See chapter 21 "Backing Up and Recovering rvsEVO Data" for information about Backup and Recovery.



Parameters for Operation and Maintenance

Parameter	Description
BackupOnStartup	Use this parameter to specify an automatic backup, performed on each time you start rvsEVO. Possible values: - `Y (default)`: each time when rvsEVO will be started the directories `$RVS_HOME/conf, /files, /system` und `/jobs` are saved in a .jar file in `$RVS_HOME/archive` directory. Also database data are saved. The name is a composition of date, time and counter in format `YYMMDD-hhmmssccc.jar` - `N`: no automatic backup
CleanupDays	Specify days for cleanup of completed or failed jobs. The `archiveJobs` program will delete any jobs older than the time specified in this parameter. Default: 7 days Note: You can set only one of cleanup parameters: **cleanupdays** or **cleanuptime**

Parameters for Operation and Maintenance

Parameter	Description
CleanupInterval	Time interval in minutes between two cleanup cycles. This parameter has no relation to the other two cleanup parameters: **CleanupDays** or **CleanupTime** Default: `1440`
CleanupTime	Specify time in the format HH:mm:ss (24 hour format) for cleanup of completed and failed jobs. Any jobs older than the time specified will be deleted. Note: You can set only one of cleanup parameters: **cleanupdays** or **cleanuptime**.
MailLocalAddress	Mail address of the sender of warnings. See also **Mail** parameter in chapter 10.8
MailSMTPHost	Hostname or IP address of the mail server if warnings are sent. See also **Mail** parameter in chapter 10.8.
MaxRevisionLogCount	Number of RevisionLog files that can be generated. Default: `1000`
MaxRevisionLogSize	Maximum file size of the RevisionLog file in lines (1 line has maximal 10 kbytes). Default: `1.000.000`
MonlogInDatabase	In this parameter you define whether the log messages should be written into the database. **Important hint:** The database table is not cleaned up by rvsEVO. (I.e. the user has to clean up the table because otherwise it would fill up.) Possible values: - `Y`: the log messages are displayed - `N`: the log messages are not displayed (default)
MonlogInDatabase-Cleanup	Only relevant if **MonlogInDatabase**=Y. In this parameter you define whether the log messages should be deleted automatically from database or not. Possible values: `NONE` (Default): log messages are not deleted automatically. `AUTO`: All log messages older than 7 days will be deleted every 24 hours, by default. This configuration can be set up via the parameters **CleanupDays**, **CleanupInterval** and **CleanupTime**.

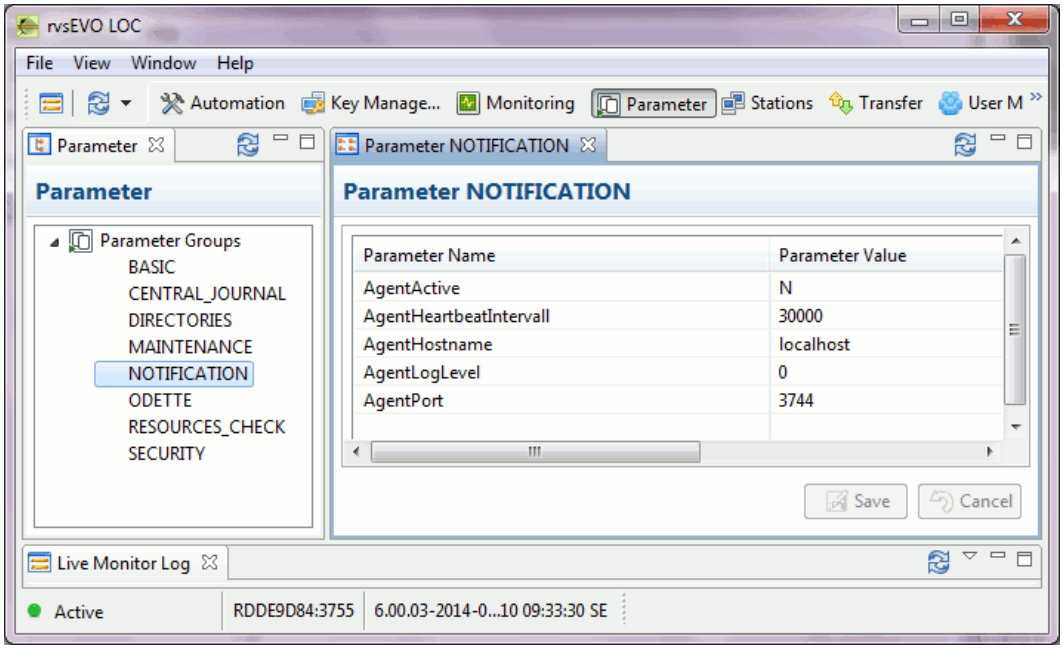
Parameters for Operation and Maintenance

Parameter	Description
PermissionTempScript	File permission of the temporary script file which is copied into temp directory after a jobstart process was started. Only if Jobstart parameter **Parameterübergabe**=REPLACE. `770` = read/write/execute for owner and group `777` = read/write/execute for all users
PersistenceArchive	Name of the file with the transfer data (statistic file). Default: `$RVS_HOME/archive/RevisionLog.xml`.
RedoLog	no function

10.6 NOTIFICATION

The parameters in this group refer to the feature `rvs® SNMP Agent`.

`rvs® SNMP Agent` is an application that can respond to Network management systems (NMS) queries and send rvsEVO status information to NMS.



You can find the installer of `rvs® SNMP Agent` in the `$RVS_HOME/SNMP_Agent` directory. For more information about this feature, please read the `rvs® SNMP Agent` User Manual. After installation of `rvs® SNMP Agent` the manual is stored in `C:/Programs/rvsSNMPAgent/doc` directory by default.

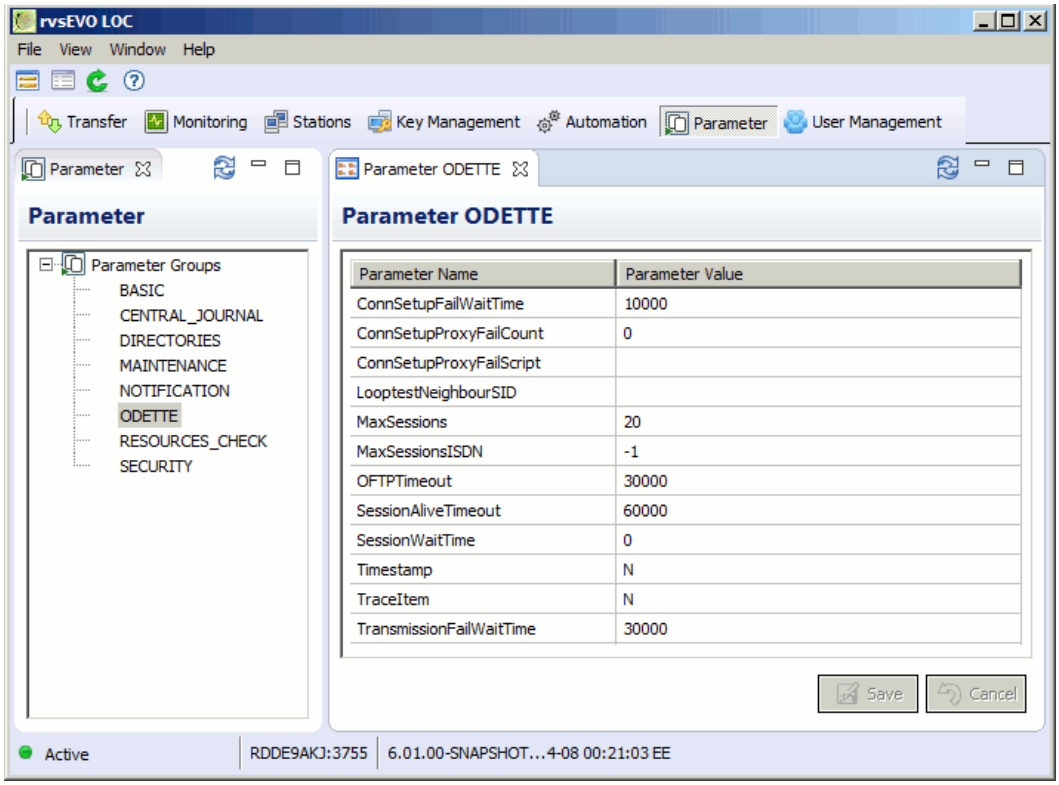
The following parameters have to be edited in rvsEVO to be able to work with rvs® SNMP Agent:

Parameters for Notification

Parameter	Description
AgentActive	This parameter defines whether or not rvs® SNMP Agent is enabled. Possible values: Y (Yes) or N (No). Default: Y (Yes)
AgentHeartbeat-Intervall	This parameter defines the interval (in seconds) at which rvsEVO sends a Heartbeat message to the Agent UDP address (see parameter **AgentHostname** and **AgentPort**). Default: 30000
AgentHostName	Agent computer name (or IP address). Default: localhost.
AgentLogLevel	This parameter defines which kind of log messages rvsEVO sends to the Agent. Possible values: - 0: no log messages are sent - 1: error messages are sent - 2: warnings and error messages are sent - 3: warnings, information and error messages are send.
AgentPort	Agent IP port. Default: 3744.

10.7 ODETTE Parameters

In this group you can set the parameters for Odette transfer.



OFTP Parameters

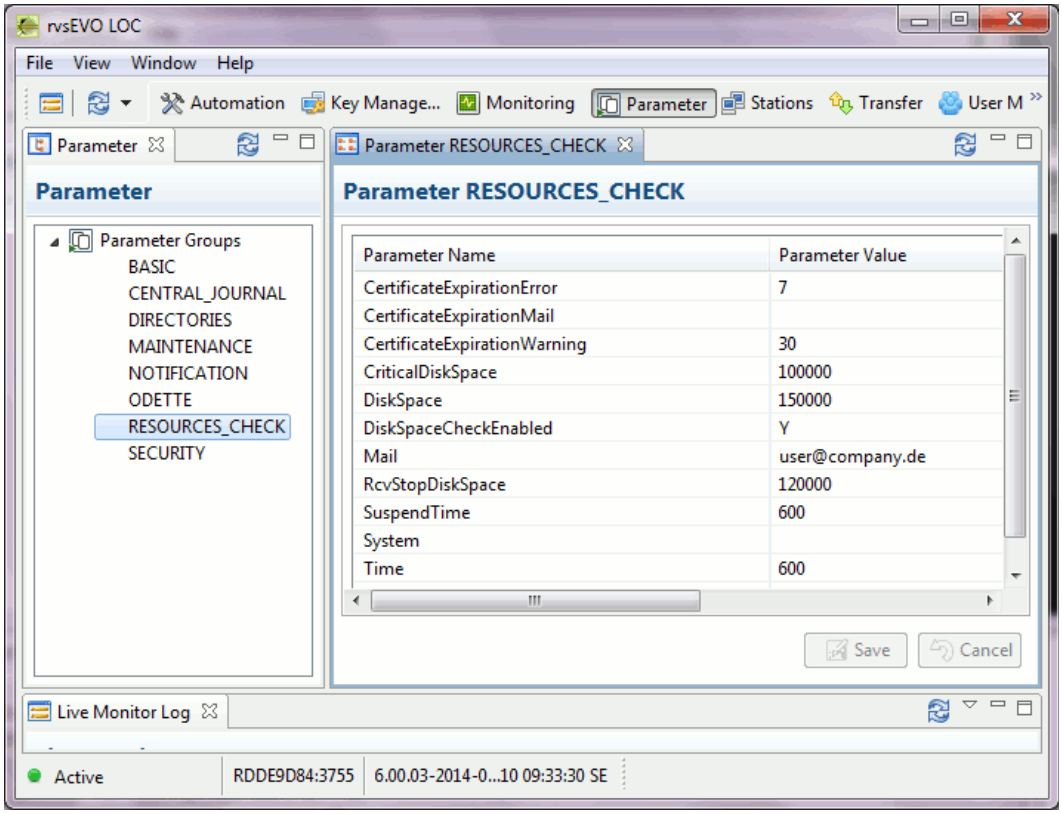
Parameter	Description
ConnSetupFailWaitTime	Time in milliseconds rvsEVO waits after a connection failed to be established before rvsEVO tries to establish the connection again. Default: `10 000`
ConnSetupProxyFail Count	Number of failed connection attempts with rvs® OFTP Proxy after which the script, defined in **ConnSetupProxyFailScript** parameter should be started.
ConnSetupProxyFail Script	Job to be started after the number of failed connections, which is defined in **ConnSetupProxy FailCount** parameter is reached.
LooptestNeighbourSID	ID of the station via which the loop test (transmission of a file to the own local or virtual station) is performed. The file is to be transmitted to the own station, and rvsEVO sends this file via a neighbour station back to the own station. If no StationID is specified the file is delivered directly to `inbox` of local station and not routed via a neighbour station.

OFTP Parameters

Parameter	Description
MaxSessions	Maximum number of simultaneously running receiving processes for TCP/IP communication. Default: `20`; Maximum number is restricted by system resources and the licensekey.
MaxSessionsISDN	Maximum number of simultaneously running receiving processes for ISDN communication. Default: `-1` (no limit) Maximum number is restricted by system resources and the licensekey.
OFTPTimeout	Time out in milliseconds at ODETTE level; default: `30 000`, no maximum
SessionAliveTimeout	Time in milliseconds to consider a connection active; Default: `600 000`, no maximum.
SessionWaitTime	**SessionWaitTime** is the time in milliseconds the OFTP session between the sender and the receiver station will be kept alive after an ended transmission for waiting for another send job. Default: `0`
Timestamp	Defines whether or not a time stamp is added to a file name name when it is received. Possible values: - `N` (default): Time stamp is added by default, only if the file name already exists; - `Y`: File name always receives the time stamp.
TraceItem	Parameter that enables tracing. The following values are possible: - `O`: for Odette level, - `N`: not active (default) See chapter 13.5 "Trace Files" for further information.
TransmissionFailWait-Time	Time in milliseconds for a transmission restart after a failure. Default: `30 000`

10.8 RESOURCES_CHECK

This parameter group refers to resource check of rvsEVO directories and checking the expiration dates of certificates. This functionality allows the operator to react early enough to lack of free disk space and expiring certificates.



Resource check

All active directories of rvsEVO will be checked: `DB, TEMP, INPUT, OUTPUT, ARCDIR und LOGDIR`.

In case of the low disk space you can see in the Monitor Log a message, that informs you in which rvsEVO directories is not enough disk space. At the same time an E-mail to the responsible administrator will be sent.

There are three levels of resource lack:

– first level: a warning will be dispayed.

– second level: all receivers will be stopped, so that no file transfer will be possible

– third level: all processes and rvsEVO itself will be ended.

Certificate check

Checking of expiration dates of your own keypairs and the certificates of your partners. If a certificate expires in preset time a message will be issued in the Monitor Log..

Parameters for Certificate and Resource Check

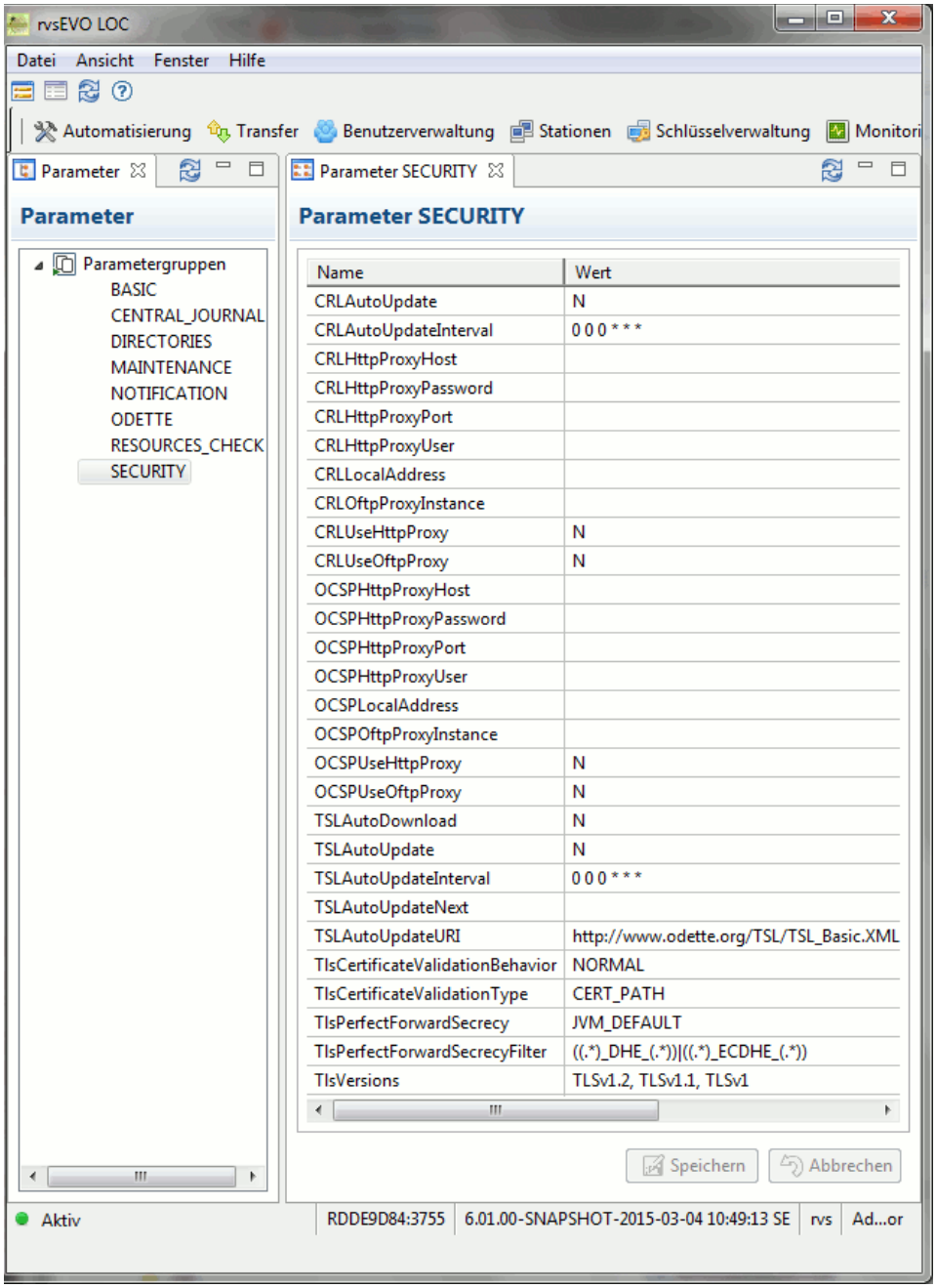
Parameter	Description
CertificateExpirationError	If certificate expiration is reached in less days than defined in this parameter an error message will be issued in Monitor Log and, if indicated, will be sent (see also parameter **CertificateExpirationMail**) Default: `7`
CertificateExpirationMail	With this parameter you define whether an email is sent in case of certificate expiration. Possible values: `Yes` / `Y`: email is to be sent `No` / `N`: email is not to be sent
CertificateExpirationWarning	If certificate expiration is reached in less days than defined in this parameter a warning will be issued in Monitor Log and, if indicated an email will be sent (see also parameter Mail). Default: `30`
CriticalDiskSpace	Number of kilobytes, that has to be free in every checking rvsEVO directory. With this parameter the critical limit for the resource deficit should be configured. If the number of free kilobytes is less than a value of this parameter all rvsEVO processes and the rvsEVO server will be stopped. You have a possibility to start a script (see parameter `System` in this table) Default: `100 000`
DiskSpace	Number of kilobytes, that has to be free in every checking rvsEVO directory, before a warning in Monitor Log will be issued. Default: `150 000` kB.
DiskSpaceCheckEnabled	With this parameter you define whether the disk space is to be checked or not. Possible values: `Y`: disk space check should be done (default). `N`: none disk space check

Parameters for Certificate and Resource Check

Parameter	Description
Mail	Email address, to which the warning is to be sent at the server startup if ... 1. the number of free disk space is less than defined in **DiskSpace** parameter. The warning will be sent cyclically as defined in **SuspendTime** parameter. 2. the values defined in parameters **CertificateExpirationWarning** and **CertificateExoirationError** are underrun. See also parameters **MailLocalAddress** and **MailSMTPHost** in “Parameters for Operation and Maintenance” on page 147
RcvStopDiskSpace	Number of kilobytes, that has to be free in every checking rvsEVO directory, before a warning in Monitor Log will be issued. If the number of free kilobytes is less than a value of this parameter all rvsEVO receiver processes will be stopped. No receiving of data will be possible. Default: `120 000` kB.
SuspendTime	Time between the resourse checks after the value given in **RcvStopDiskSpace** parameter was underrun. Default: `600`
System	a script, which should be executed, if the critical limit (level) is reached. See **CriticalDiskSpace** parameter in this table.
Time	Time in seconds between two resource checks. Default: `600`.

10.9 Security Parameters

In this group you can configure proxy settings for the monitoring of certifications (validation via CRL and OCSP). Please see chapter 22 "PKI Connection" for more information about certificate validation.


rvsEVO LOC
Datei Ansicht Fenster Hilfe
Automatisierung Transfer Benutzerverwaltung Stationen Schlüsselverwaltung Monitori
Parameter
Parameter SECURITY
Parameter
Parameter SECURITY
Parametergruppen
BASIC
CENTRAL_JOURNAL
DIRECTORIES
MAINTENANCE
NOTIFICATION
ODETTE
RESOURCES_CHECK
SECURITY
Name
Wert
CRLAutoUpdate N
CRLAutoUpdateInterval 0 0 0 * * *
CRLHttpProxyHost
CRLHttpProxyPassword
CRLHttpProxyPort
CRLHttpProxyUser
CRLLocalAddress
CRLOftpProxyInstance
CRLUseHttpProxy N
CRLUseOftpProxy N
OCSPHttpProxyHost
OCSPHttpProxyPassword
OCSPHttpProxyPort
OCSPHttpProxyUser
OCSPLocalAddress
OCSPOftpProxyInstance
OCSPUseHttpProxy N
OCSPUseOftpProxy N
TSLAutoDownload N
TSLAutoUpdate N
TSLAutoUpdateInterval 0 0 0 * * *
TSLAutoUpdateNext
TSLAutoUpdateURI http://www.odette.org/TSL/TSL_Basic.XML
TlsCertificateValidationBehavior NORMAL
TlsCertificateValidationType CERT_PATH
TlsPerfectForwardSecrecy JVM_DEFAULT
TlsPerfectForwardSecrecyFilter ((.*)_DHE_(.*))|((.*)_ECDHE_(.*))
TlsVersions TLSv1.2, TLSv1.1, TLSv1
Speichern
Abbrechen
Aktiv
RDDE9D84:3755
6.01.00-SNAPSHOT-2015-03-04 10:49:13 SE
rvs
Ad...or

.

Security Parameters

Parameter	Description
CRLAutoDownload	Only available if parameter **Certificate Validation Type**=CRL (see table “Parameters for PKI Binding” on page 265). Possible values: - Y = Yes - N = No (default) If **CRLAutoDownload**=Y an automatic download of CRL and import into the database takes place (first import into the database).
CRLAutoUpdate	Only available if parameter **Certificate Validation Type**=CRL (see table “Parameters for PKI Binding” on page 265). Possible values: - Y = Yes - N = No (default) If **CRLAutoUpdate**=Y an automatic update of CRL takes place in regular time intervals. (See also parameter **CRLAutoUpdateInterval**)
CRLAutoUpdateInterval	Time interval for automatic CRL update. A description of the format you can find under `http://quartzscheduler.org`. The whole URL you can find below the table. Exmples: „0 0 0 * * *“: The update is executed daily at 12:00 am (default). „0 15 10 ? * *“: The update is executed daily at 10:15 am. „0 0 0,6,12,18 * * ?“: The update is executed daily at 12:00 am, 6:00 am, 12:00 pm and 6:00 pm. (See also parameter **CRLAutoUpdate**)
CRLHttpProxyHost	http-Proxy server to be used for the connection to the external CRL URIs
CRLHttpProxyPassword	Password for http-Proxy in case of user/password access (for CRL download)
CRLHttpProxyPort	Port for the connection to the Http-Proxy-Server (for CRL download)
CRLHttpProxyUser	User for http-Proxy in case of user/password access (for CRL download)
CRLLocalAddress	Sender address of the CRL download request.
CRLOftpProxyInstance	Selection of a Bastion instance, configured in rvsEVO (for CRL-Download).

Security Parameters

Parameter	Description
CRLUseHttpProxy	If **CRLUseHttpProxy**=**Y** a Http-Proxy server is to be used for the CRL Download. In this case please, also consider parameters **CRLHttpProxyHost**, **CRLHttpProxyPassword**, **CRLHttpProxyPort**, **CRLHttpProxyUser** and **CRLLocalAddress**.
CRLUseOftpProxy	If **CRLUseOftpProxy=Y** an OFTP-Proxy server is to be used for the CRL Download
OCSPHttpProxyHost	http-Proxy server to be used for the OCSP check
OCSPHttpProxyPassword	Password for http-Proxy in case of user/password access (for OCSP check)
OCSPHttpProxyPort	Port for the connection to the Http-Proxy-Server (for OCSP check)
OCSPHttpProxyUser	User for http-Proxy in case of user/password access (for OCSP check)
OSCPLocalAddress	Sender address of the OCSP Requests
OSCPOftpProxyInstance	Selection of a Bastion instance, configured in rvsEVO (for OCSP check).
OSCPUseHttpProxy	If **OCSPUseHttpProxy**=Y a http-Proxy server is to be used for the OCSP check.
OSCPUseOftpProxy	If **OCSPUseOftpProxy**=Y an Oftp-Proxy server is to be used for the OCSP check.
TSLAutoUpdate	If **TSLAutoUpdate** = Y an automatic update of TSL takes place when the time in parameter **TSLAutoUpdateNext** is reached. (See also parameter **TSLAutoUpdateInterval**) Possible values: Y = Yes N = No (default)

Security Parameters

Parameter	Description
TSLAutoUpdateInterval	Time interval for checking parameter **TSLAutoUpdateNext**. A description of the format you can find under `http://quartzscheduler.org`. The whole URL you can find below the table. Exmples: „0 0 0 * * *“: The update is executed daily at 12:00 am (default). „0 15 10 ? * *“: The update is executed daily at 10:15 am. „0 0 0,6,12,18 * * ?“: The update is executed daily at 12:00 am, 6:00 am, 12:00 pm and 6:00 pm. (See also parameter **TSLAutoUpdate**)
TSLAutoUpdateNext	Time of next automatic TSL update.
TSLAutoUpdateURI	Link to TSL; default: http://www.odette.org/TSL/TSL_Basic.XML. With HTTP connection the parameters **CRLHttpProxyHost**, **CRLHttpProxyPort**, **CRLHttpProxyUser** and **CRLHttpProxyPassword** should be setted.
TlsCertificateValidationBehavior	Possible values: - `NORMAL` (default): Only the Root or CA certificate has to be imported for TLS encryption (identical to rvsEVO version 5.xx). In this case, with CRL validation the validation URI must be entered in intermediate or root certificate. **Hint:** Value `NORMAL` replaces `LEGACY`. But `LEGACY` can also be used. - `DEFAULT`: In addition to the root and server certificate the user certificate has to be imported. (See chapter 8.4.5 "Use of CA Certificates" for more information)
TlsCertificateValidationType	Only effective if parameter **TlsCertificateValidationBehavior**=`NORMAL` With this parameter the validation type can be indicated. Possible values: - `CERT_PATH` - `OCSP` - `CRL` (See chapter 22.3 "Validation Methods" for further information)

Security Parameters

Parameter	Description
TlsPerfectForwardSecrecy	Algorithms, used. Possible values: - `FORCED`: Cipher Suites List contains DHE algorithms - `CLIENT_PREFERED`: Cipher Suites List contains all algorithms, but DHE algorithmus is preferably used. - `JVM_DEFAULT`: Cipher Suites Liste remains unchanged. After changing a restart is necessary.
TlsPerfectForwardSecrecyFilter	Regular expression; Default: `((.*)_DHE_(.*))|((.*)_ECDHE_(.*))` After changing a restart is necessary.
TlsVersions	Possible values of TLS protocol: - `TLSv1` - `TLSv1.1` - `TLSv1.2` You can specify more than one version seperated by comma and spaces. Default: `TLSv1.2, TLSv1.1, TLSv1` After changing a restart is necessary.

URL for format description of **CRLAutoUpdateInterval** parameter:

`http://quartz-scheduler.org/documentation/quartz-2.1.x/tutorials/crontrigger`

10.10 Customizing via rvs-system.properties File

Below you can find the description of the of defintions of `rvs-system.properties` file.

- `rvs_evo.jobstarter.keep_temp_files=true`

  This definition refers to the jobstarts. If the value of **Parameter Handling** parameter is REPLACE the jobstarter creates temporary files including the job data which substitutes the placeholders of the script. You can define whether the temporary files are to be deleted or not. Possible values:

  `true`: the files are kept

  `false`: the files are deleted (default)

- `rvs_evo.serviceprovider.keep_temp_files=true`

  Via this definition you can define whether temporary files created by the service provider are to be deleted or not. Possible values:

  `true`: the files are kept

  `false`: the files are deleted (default)

- `rvs_evo.servicepro-vider.receive_job.process_comsecure=false`

  Via this definition you can define whether files, encrypted with ComSecure are to be received decrypted and decompressed or encrypted and compressed. Possible values:

  `true`: decryption / decompression (default)

  `false`: no decryption/no decompression

- `rvs_evo.protocol.restart=false`

  Via this definition you can define whether a restart flag is to be set up in SSID and the job is restarted.

  `true`: restart flag is set up (default)

  `false`: no restart flag

- `rvs_evo.routed_job.fsp.dont_keep.receive_file=true`

  By default, files received via `FileServiceModul` feature are kept until receiving the EERP. If `rvs_evo.routed_job.fsp.dont_keep.receive_file=true`, this function is deactivated and the files are deleted after successful routing.

- `observer.sfiddesc.usefilename=true`

  If the file transmission takes place via Observer you can define whether the original filename is to be used as file description (SFID-DESC).

- `rvs_evo.tracing.odette.enable_network_data=true`

  Via this definition you can define whether the network data is to be displayed in case of Odette tracing. Possible Values:

  `true`: network data is to display (default)

  `false`: no network data are displayed

- `rvs_evo.tracing.odette.station_ids=SID1,SID2,SID3`

  If parameter **TraceItem**=O (parameter group ODETTE of `Parameter` perspective) this definition enables Odette tracing per station.

  Values: SID for what Odette tracing is enabled. Seperate multiple station IDs by comma.

- `rvs_evo.controller.underscore_in_vdsn_allowed=true`

  Use this definition to allow underscores in VDSN. Possible values:

  `true:` underscores are admitted in VDSN

  `false:` underscores are not admitted in VDSN (default)

- `useKeyUsage=false`

  Via this definition you can define whether the certificate usage of X.509 certificate is to be considered or not.

`true`: the certificate usage will be checked and the key is used for encryption, only if Key Encipherment parameter is activated in the certificate.

`false`: the settings of X.509 certificate will not be checked.

- `javax.net.debug=ssl`

  Via this definition you can activate the debug display for TSL connections. By default, this is disabled.

- `rvs_evo.controller.check_security_by_sfid=false`

  If OFTP parameter (of partner station) **File Encryption**=forced, via this definition you can define that the setting is also valid for ComSecure. Possible values:

  `true:` The setting of **File Encryption** parameter refers to OFTP transmission only. Unencrypted files are refused with File Negative Answer (SFNA).

  `false:` The setting of **File Encryption** parameter refers to ComSecure, too. Unencrypted files are refused with End Session Indication (ESID) after receiving of the first data package.

- `java.rmi.server.hostname=<IP>`

  IP address for RMI communication.

11 User Management

Via `User Management` perspective you can create, edit or delete users and display user lists or detailed information of a user.

Hint:

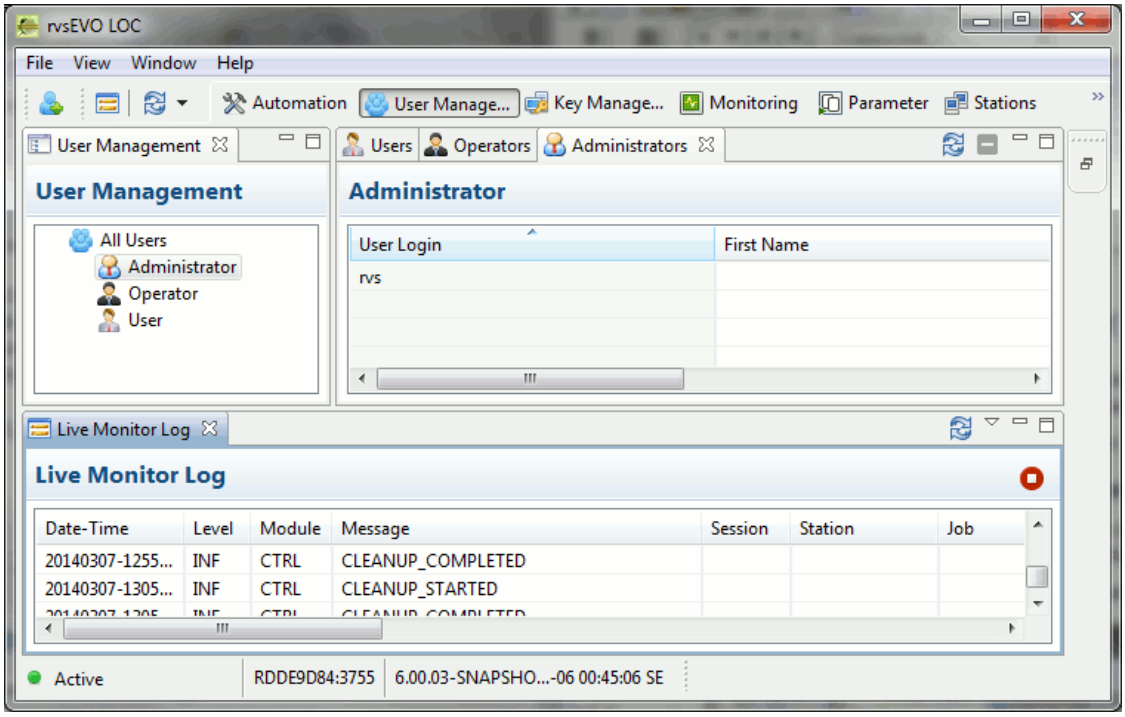
The User Management perspective is only available for user with administrator rights.

The data of default user cannot be edited or deleted.

11.1 User Interface of User Management Perspective

Select the menu item `View` in the function bar and thereafter the subentry `User Management` to open the User management perspective.

The illustration below shows the default view of Standard Edition.



The symbols in the function bar on the left side of the window provides the following functionalities:

– Show Live Monitor-Log

– Reset Perspective

– Create New User

Below, the navigation tree provides the following entries:

– All Users

  – Administrator

  – Operator

– User

At the bottom of the GUI the Live Monitor Log View is positioned.

11.2 Display Users

List view

Listings of the users can be displayed via a single-click of the user group in the navigation tree. The list will be output in a new view.

A right-click on an user in the list opens a drop-down menu which offers the following additional options `Delete User`, `Open in New View`, `Copy to Clipboard`, `Duplicate User`.

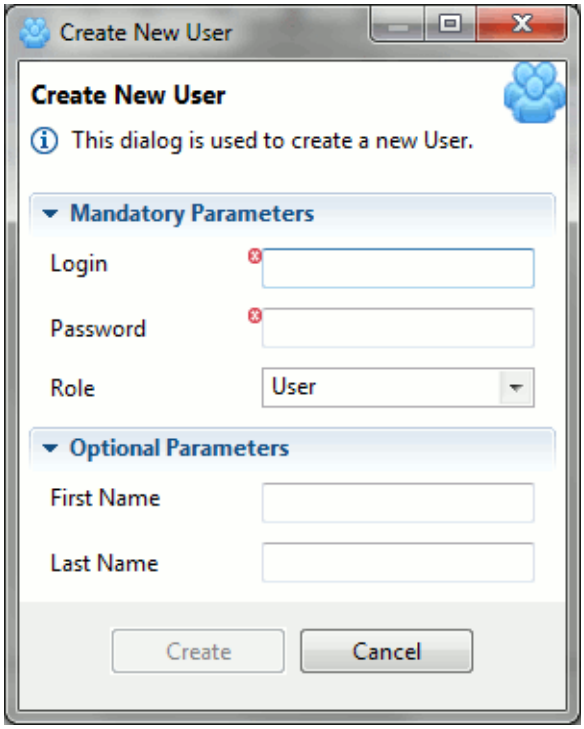
Detailed view

With a double-click on the user in the list you get detailed information about the user. Another way to output the user details is to right-click the user and thereafter select `Open in New View` from the drop-down menu. In the second case the detailed view of several jobs can be opened. The view in foreground is to be selected via tab.

11.3 Create New User

Single-click the following button in the function bar to add a user.

The dialog box `Create New User` opens.



The following parameters are mandatory:

– `Login` (user account)

– `Password` (for login)

– `Role` (Administrator, Operator or User are to be selected via combo box)

Optional parameters are the `First Name` and `Last Name` of the user.

After setting up the parameters, click the **Create** button to add the user. The last window displays the data of the new user.

Another way to add a new user is to right-click the appropriate subentry in the navigation tree and afterwards choose `Create Administator`, `Create Operator` or `Create User` from the context menu. In this case you cannot select the role in the `Create New User` dialog box.

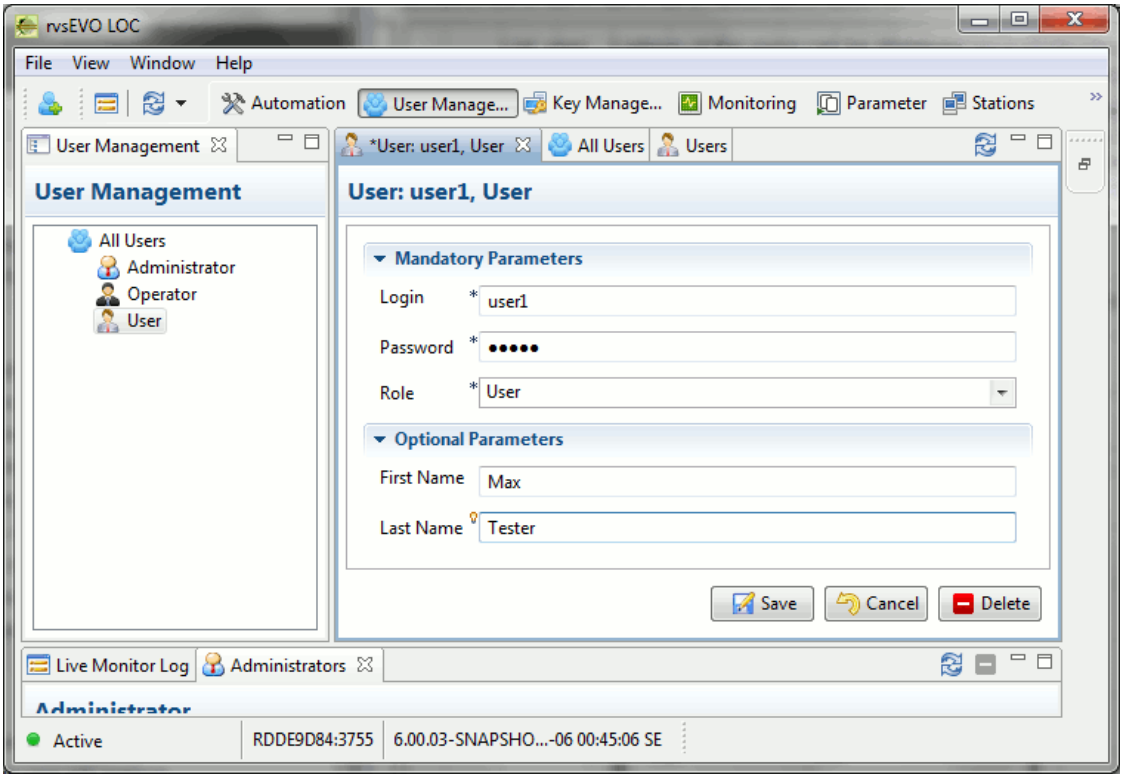
You can add an user with identical settings to an existing user. Right-click the user that is to be duplicated in the list view and select `Duplicate User` entry from the combo box. The settings of the duplicated user are used as default values of the new user. The value of **Login** parameter must be changed. The remaining parameters can also be edited if neccesary.

Hint: A basic **User** may use the rvsEVO program to send and receive files. An **Operator** may execute operator commands. An **Administrator** has the rights of the user and the operater and in addition to this he may execute the management of private keys and users and configure the rvsEVO system.

In addition to this an **Administrator** may execute the management of private keys and users and configure the rvsEVO system.

11.4 Edit User Data

Open the detailed view to edit the user data.



After making the necessary changes, press the **Save** button to confirm the modifications. If you try to open the detailed view of another user without saving the modifications, you were asked whether you want to save or discard the changes.

11.5 Delete a User

Open the detailed view of the user and press the **Delete** button or right-click the user and select Delete User from the drop-down menu. Confirm your input by pressing the **Yes** button in the next dialog box to remove the user.

Hint: You can delete several users at once by highlighting the users to be deleted (holding down the control key). Thereafter right-click one of the highlighted users and select `Delete User` or select the **`Delete`** icon in the tool bar of the list view.

12 Command Line Tools

In this chapter you get an overview and the description of the command tools in **$RVS_HOME/bin** directory.

12.1 getStationList

Use this tool to export the stationlist from the database into an XML file.

Syntax:
`getStationlist -f <File> [-help] [-?] [-verbose|-v]`

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the stationlist is to be imported.

Optional Parameters:

Parameter	Description
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

Parameter	Description
-sh <host:port>	Hostname and port for remote login.
-sp <password>	Password of the client

12.2 updateStationList

With the `updateStationList` program you have the possibility to set up the stations with a command tool at run time. For this an XML file, which contains all information about the stations you would like to modify, must be generated. The content of this file is described in a XML pattern (`$RVS_HOME/system/data/xsd/stationList.xsd`). For each station, an additional XML element (function) is necessary.

Possible directives:

– `insert`: add a new station

– `update`: modify existing stations

– `delete`: delete a station (the own station cannot be deleted)

Syntax:

```
updateStationList -f <xml filename> -r <failed
stations xml file name> -su <username> -sp <password>
```

Requested Parameters:

Parameter	Description
-f <xml file name>	Name of the XML element which contains information about the stations.
-r <failed stations xml file name>	Name of the XML file which contains the stations with failed update.

Optional Parameter:

Parameter	Description
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	Hostname and port for remote login.

12.3 File Transmission

This section contains the command tools for file transmission and display of job data.

12.3.1 activateStation

Use the `activateStation` program to activate a direct neighbour station or to activate the station and induce file transmission from this station.

Syntax:

```
activateStation -s <SID> [-p <password>] [-nt] [-
verbose][-help] [-su <user> -sp <password> -sh
<host:port>]
```

Required activateStation Parameter:

Parameter	Description
-s <SID>	Station ID to be activated

Optional parameters:

Parameter	Description
-p <password>	Set this parameter for temporary overwriting of the Odette **Send Password** in stationlist. Only this connection is set up with this password.
-nt	Set this parameter to activate the neighbour station without introducing file transmission.
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

Hint: If the activating is not successful, please notice the messages in command prompt window beginning with **Message** or **Error**.

12.3.2 createSendJob

With the `createSendJob` tool you can create a send job. This functionality is identical with the `Create New Transmission` function via GUI.

Syntax:

```
createSendJob -d <filename> -s <receiver sid> | -i
<receiver oid> -v <vdsn>
[-t <table name>] [-f <originator sid>] [-F <format>]
[-M <length>][-S <serialize> -l <label>] [-nocp] [-D
<disposition>] [-C] [-Y] [-rsr] [-sfs <set-id>] [-
sif] [-Yalg <encryption algorithmen>] [-Ycin
<issuer name>] [-Ycsn <serial no>] [-sifcin
```

```
<issuer name>] [-Ycsn <serial no>] [-help] [-?] [-
verbose] [-Sd <init date>] [-St <init time>] -[Sh] [-
idf <filename2>] [-xid <external job id>] [-so <origi-
nator sid>] [-rm <record mode>] [-su <user> -sp <pass-
word> -sh <host:port>]
```

Required createSendJob parameters:

Parameter	Description
-d <filename>	Name of the file to be send.
-s <receiver sid>	Station ID of destination
-i <receiver oid>	Odette ID of destination (as an option to Station ID)
-v <vdsn>	Virtual file name; the length of the file name used for ODETTE transfer must not exceed 26 characters. **Hint:** Via defintion `rvs_evo.controller.underscore_in_vdsn_allowed=true` in `$RVS_HOME/conf/rvs-system.properties` file you can configure that underscores are allowed in VDSN.

Optional Send Parameters:

Parameter	Description
-?	Requests help information.
-C	ODETTE compression (offline compression).
-D <disposition>	Disposition of local file after successful send attempt. Possible values: - **K** (Keep): file will not be deleted after sending - D (Delete): file will be deleted after sending.
-desc <description>	Comment, free text, available only in connection with OFTP version 2. If your partner does not support OFTP version 2, this field will be ignored.
-f <originator sid>	Odette ID of originator: local (default) or virtual station

Optional Send Parameters:

Parameter	Description
-F <format>	Format of the file to be sent: - T=text file; a stream of ASCII characters - U=unstructured (binary); byte stream - V=variable; variable record length - F=fixed; fixed record length **Hint**: set also the **-M** parameter for files in V or F format. Please see also the parameter **-rm**.
-h	Displays a description of the current command.
-idf <filename>	The ID of the job is written in this file after successful transmission (return code/error level = 0)
-j <start job>	without function
-l <label>	Name of group of serialized send jobs. User specified (descriptive) label for this job. (See parameter **Serialization**)
-M <length>	Maximal record length for the files in format 'F'or 'V'. **Hint:** with **-sfs** = 4 (OFTP 2.0) the record length will be determined automatically if **-M** is not set. Please see also the parameter **-rm <record mode>**.
-nocp	By default, creating a send entry the files to be send are copied to Outbox directory of rvsEVO. With option -nocp the original file will be deleted after transmission was successfull. (Not if the format of the file to be send is 'U') **Attention**: If you have installed a serverfarm, this parameter must not be activated. The transmitted files need to be copied to outbox directory on the NFS drive, so that each node is able to access the file. (see also chapter 15 "Multi Instance Functionality of rvsEVO").

Optional Send Parameters:

Parameter	Description
-rm <record mode>	In this parameter you indicate the mode of files in format '**F**' or '**V**'. Possible values: - TXT: text file - BIN: binary file Text File in format '**F**': If the record length deviate from the specified one (Parameter **MaxRecl**) the line will be filled with blanks or will be cut up to defined value. Text File in format '**V**': longer records will be cut up to defined value. Binary files: The length of all records is the max. record size except the length of the last one. Binary files in format '**F**': If the size of the last record is shorter than the specified one the line will be filled to the defined value.
-rsr	Active only in connection with OFTP 2.0 (CMS) (see parameter **-sfs** in this table). In OFTP 2.0 there is a possibility to request the signature of the EERP or of the NERP. Your partner must support OFTP 2.0 in this case.
-S <serialize>	The files will be sent in the same order, as the send jobs were created. The next job will only be sent, if the previous is completely finished. All send jobs for the serialization must have the same label (see parameter **-l**).
-Sd <init date>	With this parameter you can define what date the send job should be launched. Format: <yyyy-mm-dd>
-sfs <set-id>	Please see parameter **Security Feature Set** in table "Security parameters:" . Possible values: - 1 / None - 2 / ComSecure (V1) - 3 / ComSecure (V2 - 4 / OFTP 2.0 (CMS)
-Sh	With this parameter the send job is created in HOLD status (Release: please see chapter 12.3.7 "releaseJob").
-sif	File signature; only for **sfs=4** (OFTP 2.0).
-sifcsn <serial no>	Certificate serial number for file signature (only for **sfs** 4
-sifcin <issuer name>	Certificate issuer name for file signature (only for **sfs** 4)

Optional Send Parameters:

Parameter	Description
-so <originator sid>	Station ID of originator: local (default) or virtual station
-St <init time>	With this parameter you can define what time the send job should be launched. Format: <hh:mm:ss>
-t <table name>	Name of your own conversion table: For ASCII(ANSI) - EBCDIC conversion: ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273. For EBCDIC - ASCII(ANSI) conversion: IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI.
-verbose	Verbose message output.
-xid <external job id>	Parameter for an external JobID, which can refer to several rvsEVO JobIDs. The external JobID is a string of alphanumeric characters.
-Y	File will be sent encrypted
-Yalg <encryption algorithm>	See the description of the parameter **Encryption Algorithm** for GUI. Possible values: - 3DES - AES
-Ycsn <serial no>	Certificate serial number for encryption (only for **sfs** 4)
-Ycin <issuer name>	Certificate issuer name for encryption (only for **sfs** 4)

:

Optional Remote Parameters:

Parameter	Description
-rd <file name>	Remote file name of the file to be sent. The file must be in the Outbox directory
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

Examples:

```
createSendJob -d C:/text.txt -s RVS -v test
```

In this example the `C:/text.txt` file is sent to the station RVS with a virtual name `test33`. The virtual file name parameter (option `-v`) is mandatory.

```
createSendJob -d C:/text.txt -s RVS -v OFTP_TEST -F F
-M 80
```

In this example the `C:/text.txt` file is sent to the station RVS with a virtual name `OFTP_TEST`, this file is a text file of which each line has a length of 80 characters without CR/LF (`-F F -M 80`).

```
createSendJob -d C:/part.txt -s RVS -v PART  -S Y -l
AUTO
```

In this example the `C:/part.txt` file is sent to station RVS with a virtual name `PART`, this file belongs to the serialized group of files with the label (for the whole group) AUTO.

12.3.3 handleEERP: Deleting or releasing EERPs

Use the program `handleEERP` to delete or release EERPs/NERPs.

Syntax:

```
handleEERP -r|-d <JobID> [-verbose|-v] [-su <user> -sp
<password> -sh <host:port>]
```

Required Parameters:

Parameter	Description
-d <JobID>	ID of the job for which an EERP is to be deleted.
-n <JobID>	ID of the job for which an NERP is to be deleted.
-r <JobID>	ID of the job for which the EERP is to be released.

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

12.3.4 holdJob

Use `holdJob` to pause a job. The job changes to status `HOLD` until it is released again.

Syntax:

```
holdJob -n [-su <user> -sp <password> -sh <host:port>]
```

Required Parameters:

Parameter	Description
-n	The jobID of the job to be held

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.3.5 deleteJob

Use the program `deleteJob` to delete a job in `HOLD` status.

Syntax:

```
deleteJob -n<jobID> [-f] [-su <user> -sp <password>
-sh <host:port>]
```

Required Parameter:

Parameter	Description
-n	ID of the job to be deleted

Optional Parameters:

Parameter	Description
-f	This parameter allows to delete active jobs without pausing the jobs before
-verbose	Verbose message output.

Optional Parameters:

Parameter	Description
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	Host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.3.6 deleteJobsOlder: Delete Old Jobs

Use this tool to delete jobs depending on age and state. All parameters are optional. By default send jobs in WF_EERP state are deleted.

Syntax:

```
deleteJobsOlder [-m <month>] [-d <days>] [-h <hours>]
[-i <minutes>] [-s <state>] [-ds] [-dr] [-verbose]
```

Optional Parameters:

Parameter	Description
-m	Jobs older than given number of months are deleted.
-d	Jobs older than given number of days are deleted.
-h	Jobs older than given number of hours are deleted.
-i	Jobs older than given number of minutes are deleted.
-s	State of jobs to be deleted. Default: WF_EERP
-ds	Send jobs are deleted (default).
-dr	Receive jobs are deleted.
-verbose	Verbose message output.
-help	Displays a description of the current command.

Optional Parameters:

Parameter	Description
-?	Requests help information.

12.3.7 releaseJob

Use this program to release a send job in status `HOLD`.

Syntax:
```
releaseJob -n [-su <user> -sp <password> -sh
<host:port>]
```

Required Parameter:

Parameter	Description
-n	ID of the job to be released

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.3.8 restartJob

Use `restartJob` to restart a job in status `FAILED_WITH_SFNA` or `SP_FAILED`.

Syntax:
```
restartJob -n [-su <user> -sp <password> -sh
<host:port>]
```

Required Parameter:

Parameter	Description
-n	ID of the job to be restarted

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

Hint:

The following information is for encrypted transmission only:

Jobs which can not be decrypted or encrypted by the Service Provider were changed to the `SP_FAILED` status. Use the button **Error** to display the error number and the description. After bug fixing you can activate the job with the tool `restartJob`

12.3.9 terminateSession

With this tool an active session can be terminated.

Syntax:

```
terminateSession -s <session_id> [-su <user> -sp
<password> -sh <host:port>] [-help] [-?][-verbose|-v]
```

Required Parameter:

Parameter	Description
-s <session_id>	ID of the job to be terminated.

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.4 Display Job Data

12.4.1 getJoblist: Listing of all Send and Receive Jobs

Use the `getJoblist` program to list all jobs.

Syntax:

```
getJobList [-a][-e][-f][-verbose][-help]
[-su <user> -sp <password> -sh <host:port>]
```

Optional Parameters:

Parameter	Description
-a	Detailed information on jobs currently being processed (in addition display of creation date, SID of neighbour station, and VDSN).
-ac S	Additional information on original filename
-e	Information on determinated jobs
-f	Information on failed jobs
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

If you call getJobList without parameter the display contains a list of all jobs currently being processed.

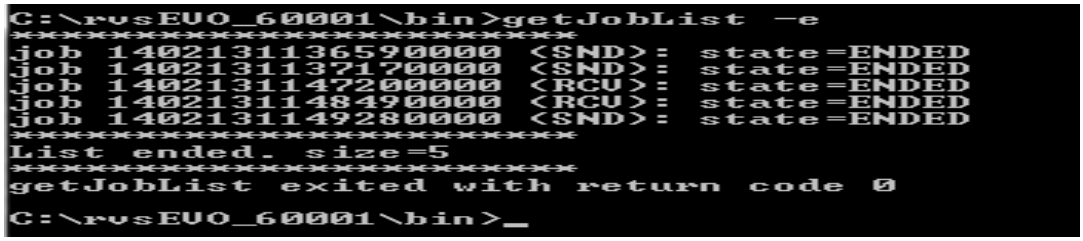
Examples:

```
getJobList -e
```

Result:


```
getJobList -a
```

Result:



12.4.2 getJob: Information on a job entry

Use `getJob` program to receive information on a particular job.

Syntax:

```
getJob -n <jobid> [-a] [-verbose|-v] [-su <user> -sp
<password> -sh <host:port>]
```

Required Parameter:

Parameter	Description
-n <jobid>	Information on a send or receive job with ID `<jobid>`.

Optional Parameters:

Parameter	Description
-a	All available job information is given.
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

Example:

```
getJob -n 040329175603000
```

Result: `job 040329175603000 (SND): state: WF_EERP`

12.4.3 getJobInfoList: Create an XML File for an External JobID

Use the program getJobInfoList to create an XML-file for an external JobID. The detailed information of all jobs which refer to the external JobID are stored in this file.

Syntax:

```
getJobInfoList -xid <external Job ID> [-f <filename>]
[-su <user> -sp <password> -sh <host:port>]
```

Required Parameter:

Parameter	Description
-xid <external Job ID>	ID of the external job. The ID can be stated with creating a transmission. (See chapter 6 "File Transmission with rvsEVO")

Optional Parameter:

Parameter	Description
-f <filename>	Filename

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

The XML schema `JobInfoList.xsd` is stored in the directory `$RVS_HOME/system/data/xsd.`

JobInfoList

The root element is `JobInfoList`. The XML element `JobExtended-Detail` contains all job information. The XML element `JobExtended-Details` contains `JobDetails` and `JobError`.

The XML element `JobError` is optional. It contains `ErrorIdentifier`, `ErrorText` and `ErrorType`.

JobInfoList: Explanation of the job attributes:

Explanation of the job attributes

Parameter	Description
compression	Compressing; possible values: true(yes) or false(no). Default: false
conversionTable	path to conversion table
creationDate	Date and time (from SFID) of creating a dispatch order in the form `DD.MM.YY HH:MM:SS`.
deleteAllowed	If **deleteAllowed**=`true` deleting a job is allowed dependent on its status. Possible values: `true` (Yes) or `false` (No). Default: `false`

Explanation of the job attributes

Parameter	Description
direction	Direction; possible values SND (sending) or RCV(receiving).
dispositiion	Whereabouts: This parameter decides whether the file to be sent remains as it is, or is deleted locally after sending. Possible values: K (Keep): File remains as it is, after sending.
encryption	Encoding; possible values: true(yes) or false(no). Default: false.
encryptionAlgorithm	Encoding algorithm; possible values: blank (no algorithm), DES_EDE3_CBC (3DES) or AES256_CBC..
encryptionCertificateIssuerName	Publisher name of the certificate. A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
encryptionCertificateSerialNumber	Serial number of the certificate. A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
errorID	If the error type is transmission, these error codes are identical with the ones from ESID, EFNA, SFNA of the Odette-log.
errorText	If the error type is transmission, these error texts are identical with the ones from ESID, EFNA, SFNA of the Odette-log.
errorType	Type of error; possible values: Internal, Transmission, MissingReceipt, Undefined.
externalJobId	JobID for external application
fileDescription	File-description. Text comment, possible after the OFTP version 2. If your destination device is not OFTP 2.0, this field is ignored.
filename	The entire path of the file, which is sent from the rvs-EVO-outbox-directory. While sending, the file is copied initially from the original-directory to the outbox-directory and then it is sent.
filenameSrc	The entire path of the original file; this file is copied in the rvsEVO-outbox-directory.
filePos	Number of the read bytes of the dispatch file.

Explanation of the job attributes

Parameter	Description
holdAllowed	This parameter gives information as to whether the stopping a job, depending on the status, is allowed. Possible values: - `true` (yes) - `false` (no).
jobNumber	rvsEVO-JobID
label	Label for serialization. All the files which have been sent in the same group, should have the same ID (label). If this parameter in case of serialization=Y(true) is not set, VDSN is used as a label.
lastByteRead	Value of the last read byte of a file to be sent in decimal representation as a character string. Default: `-1`.
lastByteSend	Value of the last sent byte of a file to be sent, in decimal representation, as a character string. Default: `-1`
lastStateChange	Time of the last change in the job status in the format `DD.MM.YY HH:MM:SS`.
lengthOriginFile	Length of the original file while sending in bytes, before processing through the service provider. This parameter is set to 0 in case of receiving.
oidDest	OdetteID of the target station.
oidNeighbor	OdetteID of the neighboring station.
oidOrig	OdetteID of the sending station (Originator)
outputCharset	without function
recCount	Number of the data records sent (records).
recordFormat	Format of the file to be transferred: - `T` (Text): a consequence of ASCII-characters - `F` (Fixed): fixed record length - `V` (variable): variable record length - `U` (unstructured): binary file.
recordLength	Maximum record length.

Explanation of the job attributes

Parameter	Description
releaseAllowed	This parameter gives information as to whether releasing a jobs depending on the status is allowed. Possible values: - `true` (yes) - `false` (no). Default: `false`.
restartPos	restart position, in case of which a new transfer of the file should be started..
scheduleDateTime	Planned time for the job start in the form `DD.MM.YY HH:MM:SS`
securityFeatureSet	Security features; Possible values: - `1 (none)` - `2 (ComSecure V1)` - `3 (ComSecure V2)` - `4 (OFTP 2.0)`
sendAttemps	Number of faulty dispatch attempts. If there is a "0", it means that this is a successful file dispatch.
serialization	This option means that your files are there in a serial order. All the files which should be sent in the same group, should have the same ID (Label). Possible values: - `true` (yes) - `false` (no). Standard value: `false`
SFIDTIME Counter	sequential counter from SFID, if several send jobs were created at the same time.
SID	SID depending on the direction: While sending: SID of the recipient; While receiving: SID of the Originator station.
sidDestination	SID of the target station
sidOriginator	SID of source station
sign	Signature; possible values: - `true` (yes) - `false` (no). Default: `false`.
signCertificateIssuer-Name	Publisher name of the signature A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.

Explanation of the job attributes

Parameter	Description
signCertificateSerialN-umber	Serial number of the signature A value is displayed, if a certificate is indicated for use explicit. It is not displayed which certificate is used in fact.
signEERP	Requesting signed EERP; possible values: - `true` (yes) - `false` (no). Default: `false`.
status	Job status. For possible status values, please see the rvsEVO-user manual (chapter 4.5 dis-patching a file).
timeStartFile	Time of the transfer receipt (system time in mil-liseconds)
transferFileLength	Sending: Size of the file to be sent. Sending: Size of the file to be sent.
transmittedBytes	Number of the actually transferred bytes
user	Number of the actually transferred bytes
VDSN	Virtual file name for Odette-transfer.
virtualSID	For future applications.
waitTime (TransmissionFailWait-Time)	Time in milliseconds, for a new start of transfer, after a failure (TransmissionFailWaitTime). Should be set in case of an SFNA(Start File Negative Answer) or EFNA (End File Negative Answer)..

```
Example:
<JobInfoList>
    <JobExtendedDetails>
        <JobDetails>
            <externalJobId>5006</externalJobId>
            <JobID>
                <direction>0</direction>
                <jobNumber>091222132218000</jobNumber>
            </JobID>
            <compression>false</compression>
```

```
            <conversionTable></conversionTable>
            <creationDate>22.12.09 13:22:18</crea-
tionDate>
            <deleteAllowed>false</deleteAllowed>
            <disposition>K</disposition>
            <direction>SND</direction>
            <encryption>false</encryption>
            <encryptionCertificateSerialNumber></
encryptionCertificateSerialNumber>
           <encryptionCertificateIssuerName></encryp-
tionCertificateIssuerName>
            <fileDescription></fileDescription>
            <filename>C:/rvsEVO/files/
outbox\rvsenv.dat.091222132218000</filename>
            <filenameSrc>C:\rvs\rvsenv.dat</filena-
meSrc>
            <filePos>789</filePos>
            <holdAllowed>false</holdAllowed>
            <lastStateChange>22.12.09 13:22:20</last-
StateChange>
            <lastByteRead>-1</lastByteRead>
            <lastByteSend>-1</lastByteSend>
            <lengthOriginFile>789</lengthOriginFile>
            <oidDest>OXP</oidDest>
            <oidNeighbor>OXP</oidNeighbor>
            <oidOrig>OGEDASEVO</oidOrig>
            <recordFormat>U</recordFormat>
            <recordLength>0</recordLength>
            <recCount>0</recCount>
            <releaseAllowed>false</releaseAllowed>
            <restartPos>0</restartPos>
            <scheduleDateTime></scheduleDateTime>
            <sendAttemps>0</sendAttemps>
            <serialisation>false</serialisation>
            <SID>XP</SID>
            <sidOriginator>LOC</sidOriginator>
```

```
            <sidDestination>XP</sidDestination>
            <signCertificateSerialNumber></signCerti-
ficateSerialNumber>
            <signCertificateIssuerName></signCertifi-
cateIssuerName>
            <status>ENDED</status>
            <timeStartFile>1261484539412</timeStart-
File>
            <transmittedBytes>789</transmittedBytes>
            <transferFileLength>789</transfer-
FileLength>
            <user></user>
            <VDSN>WEIH6644</VDSN>
            <virtualSID></virtualSID>
            <waitTime>0</waitTime>
            <jobSecurityAndSign>
                <securityFeatureSet>1</securityFea-
tureSet>
                <encryptionAlgorithm></encryptionAlgo-
rithm>
                <sign>false</sign>
                <signEERP>false</signEERP>
            </jobSecurityAndSign>
        </JobDetails>
    </JobExtendedDetails>
   </JobInfoList>
```

12.5 rvscli

This tool provides the possibility to execute the GUI functions via commandline.

Start `rvscli` via command line. After a successful start you can write the command and the parameters into the command line.

Syntax:

```
rvscli [-trace] [-help] [-su <user> -sp <password>
-sh <host:port>]
```

Optional Parameters:

Parameter	Beschreibung
-trace	This parameter activates tracing
-help **-h** **-?**	Displays a description of the current command.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

`rvscli`-Commands: Use the `help` command for listing all `rvscli` commands.

```
C:\rvsEVO5416-2\bin>rvscli
*=====================================================================*
*                                                                     *
*                 RVS Command Line Interface                          *
*                                                                     *
*=====================================================================*

                        Version:6.01.00

rvscli>help
!                       Command to Execute Operating System (OS) Commands
activateStation         Command to Activate Station.
acxDeliverCertificate   Command to Deliver ACX Certificate.
acxReplaceCertificate   Command to Replace ACX Certificate.
acxRequestCertificate   Command to Request ACX Certificate.
cleanupJobs             Command to Cleanup Jobs.
cleanupMonitorLogInDB   Command to Cleanup Monitor Log from Database.
cls                     Command to Clear Console.
createCSR               Command to Create Certificate Request.
createJobQuery          Command to Create Job Query.
createJobstart          Command to Create New Jobstart.
createObserver          Command to Create Observer.
createProxy             Command to Create Proxy Bastion.
createReceiver          Command to Create Receiver.
createSendJob           Command to Create New Job in Local Mode.
createSendJobRemote     Command to Create New Job in Remote Mode.
.
.
```

Insert ' `--help`' after a command for displayng the requested and optional parameters of the command.

Syntax:

```
activateStation --help
```

Insert the `help` command and a keyword for receiving a list of all commands which refer to this keyword. Insert `help --` and TAB key for listing the possible keywords.

`help --jobstart` displays a list of all commands which refer to a jobstart:

```
rvscli>help --jobstart
createJobstart  Command to Create New Jobstart.
deleteJobstart  Command to Delete Jobstart.
displayJobstart Command to Display Jobstart Details for
                given Jobstart ID.
listJobstarts   Command to Display Jobstart List.
updateJobstart  Command to Update existing Jobstart.

* Parameter value containing space should be enclosed with " " (for e.g. "r
i").

rvscli>
```

Insert the first characters of a command and the TAB key for listing all commands beginnig with the entered characters.

If you enter `cre` [TAB] the following output is displayed:

```
rvscli>create
createCSR          createJobQuery     createJobstart     createObserver
createProxy        createReceiver     createSendJob      createStation
createUser
rvscli>create
```

12.6 restartListener:

Use the `restartListener` tool to restart a listener.

Syntax:

```
restartlistener [-restart ¦ -start ¦ -stop] -n -l [-
help] [-?] [-verbose ¦ -v]
```

Optional Parameters:

Parameter	Description
-restart	With this parameter all listeners are restarted (default). **Hint**: Do not use parameters **-restart**, **-start**, and **-stop** together.
-start	With this parameter all listeners are started. **Hint**: Do not use parameters **-restart**, **-start**, and **-stop** together.
-stop	With this parameter all listeners are stopped. **Hint**: Do not use parameters **-restart**, **-start**, and **-stop** together.

Optional Parameters:

Parameter	Description
-n <number>	Listener number for starting or stopping the indicated listener (see network parameters of local station)
-l <line>	Network type for starting or stopping the listeners of the indicated type. Possible values: TCPIP, XOT, TLS, ISDN, REMOTE_TCP, REMOTE_TLS
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

12.7 cleanup: Deletion of Jobs and Monlog Data from the Database

Use this program to delete ended and failed transmission jobs or/and monlog data which are older than the defined date.

Syntax:

```
cleanup -d <date> [-r][-t <type>][-help] [-?]
[-verbose|-v]
```

Erforderlicher Parameter:

Parameter	Beschreibung
-d <date>	This option needs a date in the following format: `yyMMddHHmmss` or `yyMMdd`. The jobs or monlog data, which are older than the entered date will be deleted from the database.

Optionale Parameter:

Parameter	Beschreibung
-r	Remote, availables deletion during rvsEVO is running.
-t **-type**	Type of entry to be deleted. Possible values: - `JOBS` (default): data of ended or failed jobs - `MONLOG:` monlog data - `ALL`: jobs and monlog data
-help	Requests help information.
-?	Displays a description of the current command.

Optionale Parameter:

Parameter	Beschreibung
-verbose (-v)	Verbose message output.

12.8 archiveJobs: Deletion of Job Data from Database

Hint: The name of the program is historical. Up to rvsEVO version 6.00 `archiveJob` was used for archiving jobs.

Use this program to delete ended and failed transmission jobs from the database which are older than the defined date.

Syntax:
`archiveJobs -d <date>[-r][-help] [-?] [-verbose|-v]`

Required Parameter:

Parameter	Description
-d <date>	This option needs a date in the following format: `yyMMddHHmmss` or `yyMMdd`. The jobs, which are older than the date will be deleted from the database.

Optional Parameters:

Parameter	Description
-r	remote, availables deletion during rvsEVO is running.
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

12.9 archiveJob: Deletion of an Individual Job from the Database

Hint: The name of the program is historical. Up to rvsEVO version 6.00 `archiveJob` was used for archiving jobs.

Use this program to delete the data of a specific ended or failed transmission job which is older than the defined date.

Syntax:
`archiveJob -j <arg>[-help] [-?] [-verbose|-v]`

Required Parameter:

Parameter	Description
-jobID <arg> **-j <arg>**	ID of the job/jobs to be deleted. Several job ID have to be seperated by comma.

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

12.10 Key Management with rvsEVO

12.10.1 deliverCertificate

With this tool, a certain certificate for file encryption can be delivered to a partner station and imported in its database. Virtual file name for ODETTE transfer is `ODETTE_CERTIFICATE_DELIVER`.

Syntax:

```
deliverCertificate -s <stationID> [-i <keyIndex>]-r
<stationID> [-su <user>  -sp <password> -sh
<host:port>][-help] [-?]  [-verbose|-v]
```

Required Parameters:

Parameter	Description
-r <stationID>	StationID of the partner station, to which the certificate should be sent.
-s <stationID>	StationID of the station, whose certificate should be sent.

Optional Parameters:

Parameter	Description
-i <keyIndex>	Index of the certificate to be sent. If not indicated all certificates of the originator are sent. You find the index in the list view and detailed view of a certificate.
-verbose	Verbose message output.

Optional Parameters:

Parameter	Description
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.10.2 requestCertificate

With this tool, a certificate for file encryption can be requested from a partner station and an own one can be delivered to this station. Virtual file name for ODETTE transfer is `ODETTE_CERTIFICATE_REQUEST`.

Syntax:

```
requestCertificate -s <stationID> -i <s> [-su <user> -
sp <password> -sh <host:port>] [-help] [-?] [-
verbose|-v]
```

Required Parameter:

Parameter	Description
-s <stationID>	ID of the station, from which the certificate should be requested.

Optional Parameters:

Parameter	Description
-i <keyIndex>	Index of the certificate to be sent. If not indicated all certificates of the originator are sent. You find the index in the list view and detailed view of a certificate.
-o <stationID>	ID of the station whose certificate is to be sent. If not indicated the ID of local station is used.
-verbose	Verbose message output.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.10.3 replaceCertificate

With this tool, a certificate for file encryption can be sent to a partner station and replace an already existing certificate in this station. Virtual file name for ODETTE transfer is `ODETTE_CERTIFICATE_REPLACE`..

Syntax:

```
replaceCertificate -s <stationID> [-i <keyIndex>] -r
<stationID> [-su <user> -sp <password>  -sh
<host:port>] [-help] [-?][-verbose|-v]
```

Required Parameters:

Parameter	Description
-r <stationID>	StationID of the partner station, to which the separate certificate (certificates) are sent. The old certificate(s) (belonging to the individual station in case of the partner) are replaced and cancelled.
-s <stationID>	StationID of the own station, whose certificate should be sent and replaced.

Optional Parameters

Parameter	Description
-i <keyIndex>	Index of the certificate, which should be sent and replaced. If this parameter is not sent, all the available certificates are sent. Finally, in case of the partner, all the old certificates are replaced by new ones. You find the index in the list view and detailed view of a certificate.
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters:

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.10.4 exportCertificate2ComSecure

Use this tool to export a public key from rvsEVO database to ComSecure.

Syntax:

```
exportCertificate2ComSecure -sid <StationID> [-ki
<keyIndex> -cf <certificateFilename> -help -?]
```

Required Parameter:

Parameter	Description
--sid <StationID>	ID of the station whose certificate is to be exported to ComSecure.

Optional Parameters:

Parameter	Description
-ki <keyIndex>	Index of the certificate to be exported. You find the index in the list view and detailed view of a certificate.
-cf <certificateFilename>	Name of the file
-help	Displays a description of the current command.
-?	Requests help information.

12.10.5 importComSecurePublicKey

Use this tool to import a ComSecure public key into the rvsEVO database.

Syntax:

```
importComSecurePublicKey -sid <StationID> [-ckf
<comsecureKeyFilename> -dn <dname> -help -?]
```

Required Parameter:

Parameter	Description
--sid <StationID>	ID of the station to which the certificate is to be assigned.

Optional Parameters:

Parameter	Description
-ckf <keyIndex>	Name of the certificate file that contains the public ComSecure key
-dn <dname>	Clear designation of the key owner
-help **-h**	Displays a description of the current command.
-?	Requests help information.

12.10.6 getCertificateList

The tool `getCertificateList` shows the certificate list of a special stationID.

Syntax:

```
getCertificateList -sid <stationID> [-help] [-?]
```

Required Parameter:

Parameter	Description
-sid <stationID> **-stationID <stationID>**	ID of the station whose certificates should be listed.

Optional Parameters:

Parameter	Description
-help	Displays a description of the current command.
-?	Requests help information.

12.10.7 importTSL

Use this tool to import a TSL (**T**rust **S**ervice **L**ist) into the rvsEVO database.

Syntax:
`importTSL -f <TSL filename|URI> -o [-help] [-?] [-verbose|-v] [-sh <host:port>]  [-sp <password>]`

Required Parameter:

Parameter	Description
-f <TSL filename|URI>	Path of the file which includes the TSL or URI* for CRL download.
-o	Instead of parameter **-f**: Download and Import of TSL from Odette organisation* (`http://www.odette.org/TSL/TSL_OFTP2.XML`).

* Set the security parameters **CRLUseHttpProxy**, **CRLHttpProxyHost**, **CRLHttpProxyPort** before importing a TSL from the internet.

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters

Parameter	Description
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.10.8 importComSecureKeyPair

Use the program `importComSecureKeyPair` to import a ComSecure key pair from rvsXP / rvsX.

Syntax:
`importTSL -f <TSL filename> [-help] [-?] [-verbose|-v]`

Hint: Don’t use this functionality with 2048 bit keys.

Required Parameters

Parameter	Description
-pri <private key>	Name and path of private ComSecure key.

Required Parameters

Parameter	Description
-pub <public key>	Name and path of public ComSecure key (see also parameter **-x509**).
s <sid>	StationID of the station, the key pair belongs to.
-x509 <x509 filename>	Name and path of public key (instead of **-pub**, if the certificate is in X.509 format).

Optional Parameters

Parameter	Description
-help	Displays a description of the current command.
-?	Requests help information.

12.11 showCRLs

Use this tool to display a list of the imoprted **CRLs** (**C**ertificate **R**ovocation **L**ist).

Syntax:

```
showCrl [-help] [-?] [-verbose|-v]
```

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

12.12 Display Monitor Messages

12.12.1 showMonitorLog: Trace Current Monitor Messages

Use the `showMonitorLog` program to trace the current monitor messages and to analyze error messages. When starting the program, the last 25 messages are displayed.

Syntax:

```
showMonitorLog [-help] [-?] [-verbose|-v]
```

Optional Parameters:

Parameter	Description
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

Optional Remote Parameters

Parameter	Description
-su <user>	Client user for remote login.
-sp <password>	Password of the client
-sh <host:port>	host is the IP address or hostname of rvsEVO Server; port is the port of rvsEVO Server

12.12.2 showMonitorLogFile: Show Earlier Log Messsages

Use `showMonitorLogFile` to view earlier monitor log messages. The log files are saved in `$RVS_HOME/log` directory.

Syntax:

```
showMonitorLogFile -i <filename> [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-i	Path and name of the log file you would like to display.

Optional Parameters:

Parameter	Description
-help	Displays a description of the current command.
-?	Requests help information.

12.13 Command Line Tools to Start and Stop rvsEVO

The description of the following programs, you can find in chapter 3 "Start and Stop of rvsEVO":

```
startGUI
startServer
rvsEVOService
stopServer
```

12.14 Command line tools for internal use

The following programs are only for internal use:

```
login
setclientcp
setcp
userManagerClient
```

13 Setting Up of the Log and Trace Files

rvsEVO messages and warnings are saved in the log files in directory `$RVS_HOME/log`.

The `monitor.log` file contains the Monitor messages, the `rvs.log` file contains the messages from rvsEVO Server, the `rvsClient.log` file contains the messages from the command prompt and the `rvsClientRCP.log` file contains the messages of the RCP GUIs.

The trace files contains the network data.

13.1 monitor.log File

The Monitor messages are saved in the file `$RVS_HOME/log/monitor.log`. A new `monitor.log` file is generated daily and the old `monitor.log` file is renamed in „monitor.log“ plus date pattern plus counter. Also a new file is generated if the maximum file size is reached.

The maximum file size and maximum number of `monitor.log` files can be defined in the element `appender name="monlog"` in the `$RVS_HOME/conf/rvsLogger.xml` file.:

– Maximum size: use the function MaxFileSize

  Syntax:
  `<param name="MaxFileSize" value="10MB"/>`

– Maximum number of log files: use the function MaxRollFileCount

  Syntax:
  `<param name="MaxRollFileCount" value="50"/>`

Output Parameters of Log Messages:

Parameter	Description
Date-Time	Date and time of the entry (format `yyyyMMdd-HHmmss`)
Level	Kind of message. Possible values: - `INF`: Information - `WRN`: Warning / importent hint - `ERR`: ERROR

Output Parameters of Log Messages:

Parameter	Description
Module	Module of rvsEVO, which transmitted the message. Possible values: - SRV: rvsEVO Server - CTRL: controller - NET: network - OFTP: Odette protocol - CONF: configuration - PERS: persistence - ACX: automatic exchange of certificates - SP: Service Provider - OBSR: Observer
Message	Message key
Session	ID of OFTP session
Station	SID of neighbour station
Job	rvsEVO jobID
Text	Text message from message key (GUI) or parameter for text message (log file)

13.2 rvs.log File

The error messages of rvsEVO Server are saved in `$RVS_HOME/log/rvs.log` file. These messages are not displayed via GUI.

The maximum file size and maximum number of `rvs.log` files can be defined in the element `appender name="rvslog"` in the `$RVS_HOME/conf/ rvsLogger.xml` file.:

– Maximum size: use the function `maxFileSize`

Syntax:
```
<param name="MaxFileSize" value="2097152"/>
```

– Maximum number of log files: use the function `maxBackupIndex`

Syntax:
```
<param name="maxBackupIndex" value="10"/>
```

13.3 rvsClient.log File

The warnings and error messages of the command prompt are saved in `$RVS_HOME/log/rvsClient.log` file.

The maximum file size and maximum number of `rvsClient.log` files can be defined in the element `appender name="Standard"` in the `$RVS_HOME/conf/rvsClientLogger.xml` file:

– Maximum size: use the function `maxFileSize`

Syntax:
```
<param name="maxFileSize" value="2097152"/>
```

– Maximum number of log files: use the function `maxBackupIndex`

Syntax:
```
<param name="maxBackupIndex" value="10"/>
```

13.4 rvsClientRCP.log File

The warnings and error messages of the rvsEVO clients (RCP GUIs) are saved in `$RVS_HOME\log\rvsClientRCP.log` file.

The maximum file size and maximum number of `rvsClientRCP.log` files can be defined in the element `appender name="Standard"` in the `$RVS_HOME\conf\rvsClientRCPLogger.xml` file:

– Maximum size: use the function `maxFileSize`

Syntax:
```
<param name="maxFileSize" value="2097152"/>
```

– Maximum number of log files: use the function `maxBackupIndex`

Syntax:
```
<param name="maxBackupIndex" value="10"/>
```

13.5 Trace Files

The trace files contain the transmitted Odette data packages. The name is composed as follows: StationID and „_“ and SessionID and .trc extension (e.g. `RVSEVO_6372170.trc`). Trace files are saved in `$RVS_HOME/log/trace/odette` directory. Enable the tracing functionality via **TraceItem** parameter (see table “OFTP Parameters” on page 151).

14 rvsbat Batch Interface

`rvsbat` is for compability with rvs® portable. `rvsbat` gives you an additional possibility for **creating send jobs** and **managing jobstarts**.

14.1 Working with rvsbat

In order to create a send job or to manage jobstarts via rvsbat, follow the procedure given below:

- start rvsbat via command line
- enter the command for send jobs (`SEND`) or for management of jobstarts (`RESENTR`, `SENDJOB` or `FAILURE`):
  - via input file (see chapter 14.1.2) or
  - via the command prompt window (see chapter 14.1.3).

Hint: Commands and parameters are not case-sensitive.

14.1.1 Starting rvsbat

`rvsbat` tool is located in `$RVS_HOME/bin` directory and can be started via command line:

Syntax:

```
rvsbat [/c] [/i<input file>] [/q]
```

Optional rvsbat Parameters:

Parameter	Description
/i <input>	Read commands from input file. The input file may contain the following elements: - Comment lines (starting with *) - Commands (may extend over several lines by specifying **+** as the last character in the line to be continued)
/c	Continue with `rvsbat` after an error occured during execution of a utility command. By default, `rvsbat` will terminate after an error.
/q	Execute user commands in quiet mode, i.e. do not echo them to standard output; feedback about success or failure of the operation will still be provided.

14.1.2 Import a Command via Input File:

Using the example of `SEND/CREATE` command we describe what steps are neccessary to import the rvsbat commands and the parameters via input file:

- Provide the file to be transfered (example: `test.txt`)
- Provide the input file which contains the rvsbat command and send parameters.
- **Example** for an input file (`input.txt`):

  ```
  SEND /C SIDORIG=LOC DSN=C:\docs\test.txt (SID=EVO54E
  DSNNEW=TESTVDSN)
  ```

  In the above example a send job is created for sending the file `C:\docs\test.txt` from local station `LOC` to station `EVO54E` with VDSN `TESTVDSN`.
- Start `rvsbat` with the input file `input.txt`.

  Expample:

  ```
  rvsbat /Ic:\docs\input.txt
  ```

14.1.3 Call the Command via Command Prompt

To enter the rvsbat commands and the parameters into the command prompt follow the procedure given below (using the example of `SEND/CREATE` command):

- Call rvsbat via command prompt

  Example:

  ```
  rvsbat
  ```
- Input the commands and the send parameters into the prompt window after `rvsbat` was started successfully (message „Command ’`START /USER`’ successfully executed“)

  Example:

  ```
  SEND /CREATE SIDORIG=LOC DSN=C:\docs\test.txt
  (SID=EVO54E DSNNEW=TESTVDSN)
  ```

Hint: If the command is called via command prompt **rvsbat** can be terminated by **<STRG> C**.

In the following chapters you find a detailed description of the commands.

14.2 Create a Send Job with SEND/CREATE Command

Use the `SEND /CREATE` (or `SEND /C`) command of `rvsbat` script for sending files.

You can create a send job by starting rvsbat and entering the command in the command promt or reading the command from an input file (see chapter 14.1.2 and 14.1.3).

The successful transfer is confirmed in both cases via the message „Command ’`SEND /CREATE`’ successfully executed“.

Syntax:

```
SEND /CREATE DSN=<filename> [SIDORIG=<originator sid>
OIDORIG=<originator OID> CODEIN=<file code>
DISP=<disposition> FORMAT=<format> MAXRECL=<length>
LABEL=<label> SERIAL=<set id> VFTYP=<set id>
INITTIME=<set id> NOCOPY=<set id>] (SID=<receiver sid>
OIDDEST=<receiver oid> DSNNEW=<vdsn> [CODEOUT=<set id>
CODETABLE=<table name> COMPRESSION=<set id> ENCRYP-
TION=<set id> SFS=<set id> ALG=<set id>
CRYPTCERTSN=<serial number> CRYPTCERTIN <issuer name>
SIGN=<set id> SIGNCERTSN=<serial number>
SIGNCERTIN=<issuer name> SIGNRESP=<set id>
FILEDESC=<file description> XID=<external job id>])
```

14.2.1 Send /Create Parameters Supported by rvsEVO

Required Send /Create Parameters:

Parameter	Description
DSN	Path and name of local data set to be sent.
DSNNEW	Virtual file name; the length of the file name used for ODETTE transfer must not exceed 26 characters. **Hints:** The parameter is to be enclosed in parentheses. Via defintion `rvs_evo.controller.underscore_in_vdsn_allowed=true` in `$RVS_HOME/conf/rvs-system.properties` file you can configure that underscores are allowed. (GUI name: **VDSN**)
OIDDEST	Odette ID of receiver (enclosed in parantheses). Only one of the parameters must be specified: **OIDDEST** or **SID**.
SID	Station ID of receiver (enclosed in parantheses). Only one of the parameters must be specified: **OIDDEST** or **SID**.

Optional Send /Create parameters:

Parameter	Description
CODEIN	Code of local file. Possible values: - A=ASCII - E=EBCDIC Default: code of local system

Optional Send /Create parameters:

Parameter	Description
DISP	Disposition of local file after successful send attempt. Possible values: - **K**=keep; file will not be deleted after sending (default) - **D**=delete. (GUI name: **Disposition**)
FORMAT	Format of the file to be sent. Possible values: - T=text file; a stream of ASCII characters - U= unstructured (binary); byte stream - F=fixed; fixed record length - V=variable; variable record length Default: U
INITTIME	With this parameter you can decide, what time the send job is to be created. Possible values: - date and time in format "YYYY-MM-DD HH:mm:ss" - H or HOLD: job is created with HOLD status. - N or NOW: job is started immediately (default) (2 GUI parameters: **Hold, Schedule**)
LABEL	User label (up to 20 characters) used to serialize on a proceeding send job (if **SERIAL**=Y) or which can be used for serialization by a subsequent send request
MAXRECL	Max. record length for the files in format F or V (see also FORMAT parameter)
NOCOPPY	With this paramter you can decide whether the file which should be transmitted is to copy to Outbox directory or not. Possible values: - Y: no copy - N: copy is saved in Outbox dorectory (default) Note: not with format **U** and code conversion
OIDORIG	Odette ID of originator (OID of local station or OID of virtual Station)
SERIAL	If you set **SERIAL**=Y, the files will be sent in the same sequence as the send jobs were created. The next job will only be sent, if the previous is completely finished. All send jobs for the serialization must have the same **LABEL**.Possible values: - Y (Yes) - N (No) (GUI name: **Serialization**)
SIDORIG	Station ID of originator; local station (default) or virtual station. (GUI name: **SID Originator**)

Optional Send /Create parameters:

Parameter	Description
VFTYP	In this parameter you indicate the mode of files in format 'F' or 'V' (see also **FORMAT** parameter). Possible values: - X: text file (record mode 'TXT' in the GUI) - U: binary file (record mode 'BIN' in the GUI) Please see parameter **Record Mode** in chapter 6.2.1 "Send Parameters" for further information. (GUI name: **Record Mode**)

Optional Send /Create Parameters (enclosed in parentheses)

Parameter	Description
ALG	Active only in connection with OFTP 2.0 (**sfs = 4** / OFTP 2.0/CMS). The following algorithms are possible: - 3DES - AES (GUI name: **Encryption Algorithm**)
CODEOUT	Desired code of data set at receiver; possible values: - A=ASCII - E=EBCDIC Example: `send /c dsn=C:\test22.dat CODEIN=A FORMAT=V(SID=RTT CODEOUT=E DSNNEW= FIX0GBE.TEXT)`
CODETABLE	Defines the code table, which is to be used for the code conversion (see Parameter **Conversion table** in the table in "Required send parameters:" on page 81) . Indicate the alias name of conversion table. Example: `send /c dsn=c:\programs\ rvsEVO\files\outbox\test22.dat FORMAT=V(SID=RTT CODETABLE=ASCII-IBM037 MAXRECL=80 DSNNEW=FIX0GBE.TEXT).` (GUI name: **Conversion Table**)

Optional Send /Create Parameters (enclosed in parentheses)

Parameter	Description
COMPRESSION	Compression during transmission; active only in connection with **sfs**=2, 3 oder 4. Possible values: - Y (Yes): the file is to be compressed before sending. - N (No/default) This parameter has to be placed after SID parameter. Example: `SEND /C DSN=\home\test\test11.txt (SID=RTZ COMPRESSION=Y).` (GUI name: **Offline Compression**)
CRYPTCERTSN	Serial number of the certificate for file encryption; active only in connction with **sfs**=4 and **ENCRYPTION**=Y (GUI name: **Encryption Cert. Serial Number**)
CRYPTCERTIN	Issuer name of the certificate for file encryption; active only in connction with **sfs**=4 and **ENCRYPTION**=Y (GUI name: **Encryption Cert. Issuer Name**)
ENCRYPTION	File encryption; active only in connection with **sfs**=2, 3 or 4; Possible values: - Y (Yes): the file is to be encrypted before sending. - N (No): None encryption. This parameter has to be placed after SID parameter. Example: `SEND /C DSN=\home\test\test11.txt (SID=RTZ COMPRESSION=Y ENCRYPTION=Y).`
FILEDESC	File description, active only with **sfs**=4 (OFTP 2.0) (GUI name: **File description)**
NOCOPY	With this parameter you can decide whether the file which should be transmitted is to copy to `Outbox` directory or not. **Attention:** If you have installed a serverfarm, this parameter must not be activated. The transmitted files need to be copied to `outbox` directory on the NFS drive, so that each node is able to access the file. (see also chapter 15 "Multi Instance Functionality of rvsEVO").
SFS	**S**ecurity **F**eature **S**et; this parameter applies to the format of encryption. Possible values: - 1 (no encryption; default) - 2 (ComSecure V1) - 3 (ComSecure V2) - 4 (OFTP 2.0 (CMS)) (GUI name: **Security Feature Set**)

Optional Send /Create Parameters (enclosed in parentheses)

Parameter	Description
SIGN	File signature is activated. Active only in connection with **sfs**=4 (OFTP 2.0). Possible values: - Y: File Siganture is active - N: None file signature (default) (GUI name: **File Signature**)
SIGNCERTSN	Serial number of the certificate for file signature; active only in connction with **sfs**=4 and **SIGN**=Y (GUI name: **Encryption Cert. Serial Number**)
SIGNCERTIN	Issuer name of the certificate for file signature; active only in connction with **sfs**=4 and **SIGN**=Y (GUI name: **Encryption Cert. Issuer Name**)
SIGNRESP	Request signed EERP/NERP. Active only in connection with **sfs**=4 (OFTP 2.0). Possible values: - Y: signed EERP/NERP is requested - N: signed EERP/NERP is not requested (default) (GUI name: **Request Signed EERP/NERP**)
TSTAMP	No Function; for compatibility with rvs® portable. Possible values: - Y - N (default
XID	Parameter for an external JobID, which can refer to several rvsEVO JobIDs. The external JobID is a string of alphanumeric characters. (GUI name: **External Job ID**)

14.3 Managing of Jobstarts via rvsbat

In this Chapter you get information about managing of jobstarts via `rvsbat`. Jobstarts after receive (`RESENTR` command), after send attempt (`SENDJOB` command) and after processing error (`FAILURE` command) can be created, modified or deleted. Please read chapter 9.2.1 "Creating a Jobstart" for further information about jobstarts.

You can create, update or delete a jobstart by starting `rvsbat` and thereafter entering the appropriate command in the command promt or by reading the command from an input file (see chapter 14.1.2 and 14.1.3).

14.3.1 Jobstart after Receive

You can use the `RESENTR` command of `rvsbat` script for managing Jobstarts after receive (in accordance with resident receive entries in rvs® portable).

The function that is to be executed can be indicated with the RESENTR command:

– RESENTR /CREATE or /C: create a jobstart after receive

– RESENTR /UPDATE or /U: edit a jobstart after receive

– RESENTR /DELETE or /D: delete a jobstart after receive

Hint: The commands are not case-sensitive. They can be added in abbreviated or long version and with blank or without. (e.g. RESENTR /DELETE DSN="*.TENNIS.*" SID="*" is the same as RESENTR/D DSN="*.TENNIS.*" SID="*").

Syntax:

```
RESENTR /CREATE DSN=<filename> SID=<originator sid>
[CODETABLE=<table name> CODETRANS=<set id>
COMMENT=<text> DSNNEW=<new dsn name> ENABLED=<set id>
EXECSYNC=<set id> EXECTIMEOUT=<set id> JOB=<scrip
name> NEWDIR=<new directory name> NEWNAME=<new file
name> PARAMHANDLING=<set id> REPLACE=<set id>
SHELL=<set id> SIDDEST <receiver sid> TSTAMP=<set id>
TSTAMPFORMAT=<set id> VFTYP=<set id> DISP=<set id>
ACCOUNT=<set id> FLAGCOMP=<set id> FLAGCRYP=N
LUID=<set id> UID=<set id>)
```

RESENTR parameters supported by rvsRVO:

Required parameters: DSN, SID oder SIDORIG

Optional parameters: CODETABLE, CODETRANS, COMMENT, DISP, DSNNEW, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, NEWDIR, NEWNAME, PARAMHANDLING, REPLACE, SHELL, SIDDEST, TSTAMP, TSTAMPFORMAT, VFTYP

Parameters without function (for compatibility with rvs® portable): ACCOUNT, FLAGCOMP, FLAGCRYP, LUID, UID

In chapter 14.3.4 "Jobstart Parameters" you can find a description of the jobstart parameters.

CREATE **Example for creating a jobstart after receive:**

```
RESENTR /c SIDORIG=* DSN=*TESTFILE1* SIDDEST=LOC
DSNNEW=C:\Daten-rvsbat-tests\DSNNEW_TEST.txt
ENABLED=Y LUID=* UID=*
```

Files, receiving by local station LOC with a virtual filename that includes the character set "TESTFILE1" are renamed in DSNNEW_TEST.txt and saved in C:\Daten-rvsbat-tests directory. The parameters **LUID** and **UID** are without function.

Hint: One of the parameters **SID** (=**SIDORIG**), **DSN** and **SIDDEST** has to distinguish from existing jobstarts after receive.

Update **Example for updating a jobstart after receive:**

```
RESENTR /u SIDORIG=* DSN=*TESTFILE1* SIDDEST=LOC
DSNNEW=C:\Daten-rvsbat-tests\DSNNEW_TEST.txt
REPLACE=Y ENABLED=Y LUID=* UID=*
```

In example above the previously created jobstart is updated as follows: Existing files with the same name like the new one are replaced (REPLACE=Y). By default the existing file is kept and a timestamp is added to the name of new file.

Hint: The parameters **SID** (=**SIDORIG**) and **DSN** must be identical to the jobstart that is to be updated.

DELETE **Example for deletion of a jobstart after receive:**

```
RESENTR /DELETE SIDORIG=* DSN=*TESTFILE1*
```

In example obove the previously created jobstart is deleted.

Hint: The parameters **SID** (=**SIDORIG**) and **DSN** are requested.

14.3.2 Jobstart after Send Attampt

Use the SENDJOB command to create, edit or delete a jobstart after send attempt. The function that is to be executed can be indicated with the SENDJOB command:

Commands:

– `SENDJOB /CREATE or /C:` create a jobstart after send attempt

– `SENDJOB /UPDATE or /U:` edit a jobstart after send attempt

– `SENDJOB /DELETE or /D:` delete a jobstart after send attempt

Syntax:

```
SENDJOB /CREATE VDSN=<filename> SID=<receiver sid>
ATTEMPTS=<set id> COMMENT=<text> ENABLED=<set id>
EXECSYNC=<set id> EXECTIMEOUT=<set id> JOB=<scrip
name> PARAMHANDLING=<set id> SHELL=<set id>
SIDSENDER=<originator sid>  SIDORIG=<originator sid>
UID=<set id>
```

SENDJOB parameters supported by rvsRVO:

Required parameters: SID (= SIDDEST), VDSN, ATTEMPTS

Optional parameters: COMMENT, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, PARAMHANDLING, SHELL, SIDSENDER = SIDORIG

Parameters without function (for compatibility with rvs® portable):

UID, COMPRESSION; ENCRYPTION

The table in chapter 14.3.4 "Jobstart Parameters" shows the description of the parameters.

CREATE **Example for creating a jobstart after send attempt**

```
SENDJOB /c SIDDEST=EVO54 SIDORIG=LOC VDSN=*TESTF*
ATTEMPTS=0 JOB=C:\rvsTest\scripts\sendjob.bat
ENABLED=Y
```

After successful transmission (**ATTEMPTS**=0) of a file with a virtual filename that includes the character set `"TESTF"` from station `LOC` to station `EVO54` the job `C:\rvsTest\scripts\sendjob.bat` is started.

Hint: One of the parameters **SID** (=**SIDDEST**), **VDSN** and **ATTEMPT** has to distinguish from existing jobstarts after send attempt.

Update **Example for updating a jobstart after send attempt:**

```
SENDJOB /u SIDDEST=EVO54 SIDORIG=LOC VDSN=*TESTF*
ATTEMPTS=0 JOB=C:\rvsTest\scripts\sendjob.bat
ENABLED=N
```

In this example the priviously created jobstart after send attempt is disabled (**ENABLED**=`N`).

Hint: The parameters **SID** (=**SIDDEST**), **VDSN**, **ATTEMPTS** and **SIDORIG** must be identical to the jobstart that is to be updated. (**SIDORIG** only if indicated with creation).

Delete **Example for deletion of a jobstart after send attempt:**

```
SENDJOB /d SID=EVO54 SIDORIG=LOC VDSN=*TESTF*
ATTEMPTS=0
```

The previously created jobstart after send attempt is deleted.

Hint: The parameters **VDSN, ATTEMPTS, SIDDEST** and **SIDORIG** are requested. (**SIDORIG** only if indicated with creation).

14.3.3 Jobstart after Failure

Use the `FAILURE` command to create, edit or delete a jobstart after processing error. The function that is to be executed can be indicated with the `FAILURE` command:

Commands:

– `FAILURE /CREATE or /C:` create a jobstart after processing error

– `FAILURE /UPDATE or /U:` edit a jobstart after processing error

– `FAILURE /DELETE or /D:` delete a jobstart after processing error

Syntax:

```
FAILURE /Create VDSN=<filename> SIDDEST=<receiver sid>
SIDORIG=<originator sid> COMMENT=<text> ENABLED=<set
id> EXECSYNC=<set id> EXECTIMEOUT=<set id> JOB=<scrip
name> PARAMHANDLING=<set id> SHELL=<set id>
```

FAILURE parameters supported by rvsRVO:

Required Parameter: VDSN

Optional parameters: COMMENT, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, PARAMHANDLING, SHELL, SIDDEST, SIDORIG

The table in chapter 14.3.4 "Jobstart Parameters" shows a description of the parameters.

CREATE **Example for creation a jobstart after failure:**

```
FAILURE/C VDSN=Test.txt SIDORIG=LOC SIDDEST=RVS
JOB=C:\rvsTest\scripts\failure.bat
```

After receiving an error massage after send attempt of a file with VDSN `"Test.txt"` from station `LOC` to station `RVS` the job `"C:\rvsTest\scripts\failure.bat"` is to be started.

Hint: The parameter **VDSN** has to distinguish from existing jobstarts after failure.

UPDATE **Example for updating a jobstart after failure:**

```
FAILURE/U VDSN=Test.txt SIDORIG=LOC SIDDEST=RVS
JOB=C:\rvsTest\scripts\failure.bat ENABLED=N
```

In the example above the priviously created jobstart is disabled (**ENABLED**=Y)

Hint: The parameters **VDSN**, **SIDORIG** and **SIDDEST** must be identical to the jobstart that is to be updated. (**SIDORIG** and **SIDDEST** only if indicated with creation).

DELETE **Example for deletion a jobstart after failure:**

```
FAILURE/D VDSN=Test.txt SIDORIG=LOC SIDDEST=RVS
```

In the example above the priviously created jobstart after failure is deleted.

Hint: The parameters **VDSN**, **SIDORIG** and **SIDDEST** are requested. (**SIDORIG** and **SIDDEST** only if indicated with creation).

14.3.4 Jobstart Parameters

The following table shows the description of the jobstart parameters.

Required Jobstart Parameters:

Parameter	Description
DSN	(only `RESENTR`) Virtual file name. Allowed: SID string and wildcard '*****' at the end, the beginning or instead of the string. (GUI name: **VDSN**)
SID **or SIDDEST**	(only `SENDJOB`) Station ID of target station. Allowed: SID string and wildcard '*****' at the end. (GUI name: **SID of Destination**)
SID	(only `RESENTR`) Station ID of source station. Allowed: SID string and wildcard '*****' at the end. (GUI name: **SID of Destination**).

Required Jobstart Parameters:

Parameter	Description
VDSN	(`SENDJOB` and `FAILURE`) Path and name of the local file to send. Allowed: SID string and wildcard '*****' at the end, the beginning or instead of the string.

Optional Jobstart Parameters

Parameter	Description
ACCOUNT	without function; for compability with rvs® portable
ATTEMPTS	Number of unsuccessful send attempts. Successful file transmission is indicated by "0". (GUI name: **Send Attempts**)
CODETABLE	Alias of code table that is to be used if parameter **CODETRANS = '**T'. See parameter **Conversion table** in table “Jobfilter elements” on page 127 Example: `RESENTR /C DSN=C:\files\test22.dat CODETRANS=T CODETABLE=ASCII-IBM037 DSNNEW=TEST22`
CODETRANS	Defines whether a code translation has to be executed. Possible values: - no indication: no code translation - `E`: EBCDIC - ASCII conversion - `A`: ASCII - EBCDIC conversion - T: your own conversion table **,** that is indicated in **CODETABLE** parameter. (Please see also section “How to add your own conversion table:” on page 133)
COMMENT	free text;
COMPRESSION	no function; for compatibility with rvs® portable, possible values: - `N` - empty string
DISP	no function; for compatibility with rvs® portable, possible value: **K**

Optional Jobstart Parameters

Parameter	Description
DSNNEW	With this parameter you can save the received file with another name and path. - no indication: the file is saved with current name in inbox directory - indication of new path and name: the file is saved in the indicated directory with the new name
ENABLED	With this parameter you decide whether the job is to be started or not. Possible values: - Y (Yes): the job is started (default) - N (No): the job is not started
ENCRYPTION	no function; for compatibility with rvs® portable, possible values: N
EXECSYNC	With this parameter you indicate if the OFTP session is kept alive until the process is finished. Possible values: - y (session is kept alive) - N (Default) (GUI name: **Synchronized**)
EXECTIMEOUT	If setting **EXECSYNC** = Y: Time Out after the connection is closed by the communication program. (GUI name: **Timeout (Sync.)**)
FLAGCOMP	no function; for compatibility with rvs® portable; Possible values: - N - empty string
FLAGCRYP	no function; for compatibility with rvs® portable; Possible value: - N - empty string
JOB	Program to be started if all filter conditions apply. A defined set of parameters is transmitted to the program. (GUI name: **Process**)
LUID	no function; for compatibility with rvs® portable; Possible value: - * - empty string

Optional Jobstart Parameters

Parameter	Description
NEWDIR	With this parameter you can save the received file in another directory than Inbox directory; - no indication: file is saved in Inbox - indication of new path: the file is saved in the indicated directory (GU name: **New Directory**)
NEWNAME	With this parameter you can save the received file with another name. - no indication: VDSN is file name - indication of new name: the file is saved with the new name (GUI name: **New Filename)**
PARAMHANDLING	With this parameter you can decide how to transmit the job data to the process. Possible values: - ARGS: job data are passed as arguments (default) - ENV: job data are set as environment variables - REPLACE: for compatibility with rvs® portable. For further information see chapter “Parameter Handling” on page 130.
REPLACE	With **REPLACE** parameter you decide how to handle receiving files which own the same name like existing files. Possible values: - R or Y (yes): replace the existing file - N (no): create new data set with uniquename; timestamp is added (default)
SHELL	Command Shell for executions of the program, inicated with **JOB** parameter e.g. ksh, csh, ... on Unix systems
SIDDEST	Station ID of target station. (GUI name: **SID of Destination**)
SIDORIG	Station ID of source station (local or virtual station) (GUI name: **SID of Originator**)
TSTAMP	With this parameter you define rules for timestamp creation. Possible values: - Y = timestamp is to be generated generally - N = timestamp is to be added only if necessary (default) (GUI name: **Timestamp**)

Optional Jobstart Parameters

Parameter	Description
TSTAMPFORMAT	Format of the timestamp. Possible values: - `TIME`: time of creation of the job and counter; format: `hhmmssccc` (default) - `DATETIME`: date and time of creation of the job and counter; format: `YYMMDDhhmmssccc` - `SFID_DATETIME`: Odette timestamp of SFID; date, time and counter in format `YYMMDDhhmmsscccc` - `COUNTER`: `000000 - 999999`. If the counter is not sequential, the gap is filled first before the counter is counted up. (GUI name: **Timestamp format**)
UID	No function; for compatibility with rvs® portable; Possible value: empty string, '*****'
VFTYP	With this parameter you decide, whether each line of a received file in F or V format is terminated by line feed. Possible values: T: Files in F or V format are saved in text mode. Line feed is inserted at the end of each record (default) V: Files in V format are saved in text mode; files in F format are saved in binary mode. (GUI name: **Record Mode**)

15 Multi Instance Functionality of rvsEVO

On Linux systems rvsEVO Version 6.00 upwards provides the Multi Instance Functionality with a serverfarm.

The benefits of multi instance capability are fail-safe stability, great transmission capacity, scalability and load distribution.

A server farm comprises an array of several computers (rvsEVO servers). In case of a server failure another server can assume the tasks of a failed server.

15.1 Introduction

– The serverfarm offers significantly higher fail-safety and transmission capacity than single instance.

– A serverfarm comprises several rvsEVO servers aiming at ensuring high system availability.

– Jobs to be processed are evenly distributed among all rvsEVO servers within the serverfarm (load balancing).

– Transmission capacity can be increased or decreased by adding or removing rvsEVO servers (scalability).

– To ensure trouble-free rvsEVO operation another server can assume the tasks of a failed server.

– With regard to communication partners and operation (e.g. file transfer), rvsEVO multi instance functionalty behaves like a single rvsEVO.

15.1.1 Fail-Safety

Failure of a node

Failure of a single node causes a restart of the aborted jobs on another server instance. A node failure has not any impact on the overall system.

A failure of the entire system causes a restart of jobs combined with the restart of the rvsEVO systems.

Failure of the database

A temporary failure: the connection is set up automatically such as configured in the Connection Pool (see Chapter 19.3 "Oracle", section Connection Pooling).

If all connection attempts failed an error message is output, all activities are terminated in a controlled manner and the node or the rvsEVO system is stopped. The activites can be restarted.

Failure of the network file systems

If the read or write access failed the procedure is equivalent to the failure of the database: All activities are terminated in a controlled manner and the node or the rvsEVO system is stopped. The activities can be restarted.

15.1.2 Load Balancing

As every node in rvsEVO can assume all tasks, it is possible to evenly distribute the load among all nodes. A server with low load will start processing a new job earlier than a server with a high load because it will access the list with upcoming tasks earlier to check for new jobs which are to be processed (first come first serve).

15.1.3 Scalability

You can use the serverfarm to add new rvsEVO nodes or to remove old ones. This quantitative node scalability allows the serverfarm to process a significantly greater amount of data and jobs than a single instance. Adding or removing nodes at runtime allows the processing capacity to be dynamically adapted without having to stop rvsEVO.

15.2 System Requirements / Architecture of a Serverfarm

System requirements

To successfully operate the serverfarm you need the following components:

– Operating systems: UNIX (currently only Linux)

– Database: Oracle 11

– IP load balancer

– network file System/network drive (e.g. NFS)

Hint: The system clock of all nodes of the serverfarm is to be synchronized via NTP.

System architecture

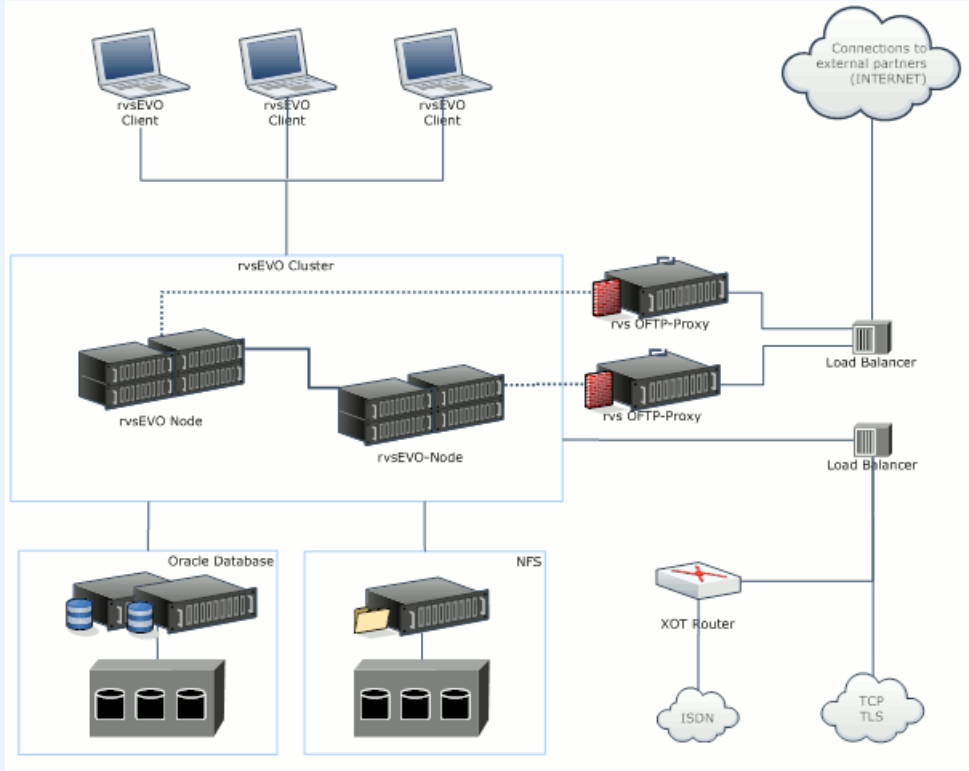
Each rvsEVO node is equipped with all functions and needs access to all resources in the cluster systems (resource sharing).

An rvsEVO cluster comprises the following components:

– several rvsEVO nodes (rvsEVO server with rvsEVO client and Oracle client).

– Network file system

The illustration below shows the system architecture of the operation with multi instance functionality.



15.3 Installation of a rvsEVO Serverfarm

Installation of a server node runs nearly analog to an installation of a single instance (see chapter 2.3 "New Installation of rvsEVO"). Please notice the following points:

- Install the Enterprise Edition of rvsEVO
- Use the Oracle database
- Insert in **IP address or hostname** parameter of server settings window the same alias name with the installation of each node.
- Define the group name with the installation of the first node in the window `Configure cluster node`. Use the same group name for each node within the serverfarm.
- Insert the host names or IP addresses of the other nodes within the serverfarm via **Nodes** parameter in the same window, multiple entries seperated by commas.
- Initialize the database only with the installation of the first node.

After installation:

- Save the directory `$RVS_HOME/files` and its subdirectories on the NFS drive. The path of the subdirectories `/inbox, /outbox und /temp` are to be set up in DIRECTORY group.

- Create directories for jobstart scripts and Observer on the NFS drive and provide the read and write access to all nodes.

Setting up of a Serverfarm

During installation the cluster node has been configured. You can set up the configuration in `$RVS_HOME/conf/cluster.properties` file. This is required after adding or removing nodes. In this case on each node of the server farm the IP addresses or host names of the added or removed nodes has to be added or deleted in `cluster.nodes` definition.

Example of cluster.properties file:

```
cluster.group.name=rvsEVO6_Cluster
cluster.node.name=rvsEVO_Node
cluster.node.id=0
cluster.node.ip=10.177.191.220
cluster.node.port=5900
cluster.nodes=10.177.xxx.1,10.177.xxx.2,10.177.xxx.3
```

Definitions of Cluster.Properties

Definition	Description
`cluster.group.name`	Name of the serverfarm (has to be identical on each node)
`cluster.node.name`	Name of the node
`cluster.node.id`	Unique node identifier. Possible values: `0-9` e.g. first node: `cluster.node.id` = `0`; second node: `cluster.node.id` = `1`; etc. You can also assign value `1` to the first node; value 2 to the second node, etc. (Default is `0`). This value is part of the job ID (fourth last position).
`cluster.node.ip`	IP address of the node
`cluster.node.port`	Port of the node (default: 5900)
`cluster.node`	host names or IP addresses of the other nodes within the serverfarm, seperated by commas

15.4 Update of rvsEVO Serverfarm

Update of a server node runs nearly analog to an update of a single instance (see chapter 2.4 "rvsEVO update installation"). Please notice the following points:

– Each node of the serverfarm has to be updated seperately.

- Before updateing you need not stop the server to be updated.
- Initialize the database with updating the last node
- Do not start the rvsEVO server before the last node was updated.

16 ODETTE Protocol

It is the purpose of the ODETTE File Transfer Protocol (OFTP) to ensure the reliable transfer of a data set. The OFTP enters a protocol session with the OFTP on the remote rvsEVO station which logically runs on top of the linedriver connection.

After the OFTP session has started, both sides exchange their ODETTE IDs and passwords, negotiate some parameters, like ODETTE exchange buffer size, ODETTE credit value (the number of buffers the sending side can send without waiting for a response), and exchange information about name, approximate size and format of the data set to be transferred.

During transfer, a compression and decompression of data is performed. After the data have been transferred, the byte count is checked between both sides. After the data set has successfully been stored, a receipt is sent to the sending station. If the transfer has been disrupted, for example by a link failure, the OFTP protocol provides a mechanism that allows to restart the transfer at the point of rupture.

'Change direction' feature allows the receiver to become sender and to send acknowledgments and data sets.

For protocol details kindly refer to the publications of the ODETTE and VDA groups: "ODETTE Specifications for File Transfer".

Shown below is the general but simplified message flow within an ODETTE session. The sending side acts as initiator, the receiving side as responder.

Initiator (sender)		Responder (receiver)
Connect	----- network connect ----->	
	<----- SSRM -----	„ODETTE FTP READY“
Odette ID, password. buffersize, credit	----- SSID ----->	
	<----- SSID -----	Odette ID, password etc.
Start**F**ile **ID** (name, size, format)	----- SFID ----->	
	<----- SFPA -----	**S**tart**F**ile **P**ositive **A**nswer
’n’ data records	----- DATA -----> ----- DATA -----> ----- DATA -----> ...	

	<----- CREDIT -----	send credit value ’n’
send ’n’ data records	----- DATA -----> ----- DATA -----> . .	
End**F**ile **ID** (byte count	----- EFID ----->	
	<----- EFPA ----- or <----- EFNA -----	**E**nd**F**ile **P**ositive **A**nswer (if store successful and byte count correct)/ **E**nd**F**ile **N**egative **A**nswer
Change **D**irection	------ CD ------>	
	<----- EERP ----- or <----- NERP -----	**E**nd-to-**E**nd **R**es**p**onse/ **N**egative **E**nd-to-End **R**es**p**onse (acknowledge)
	<----- ESID -----	**E**nd-**S**ession **ID**
Network disconnect		Network disconnect

17 rvs® OFTP Proxy

This chapter describes the configuration and the functionality of rvs® OFTP Proxy implementation in rvsEVO.

17.1 Basics

Internet

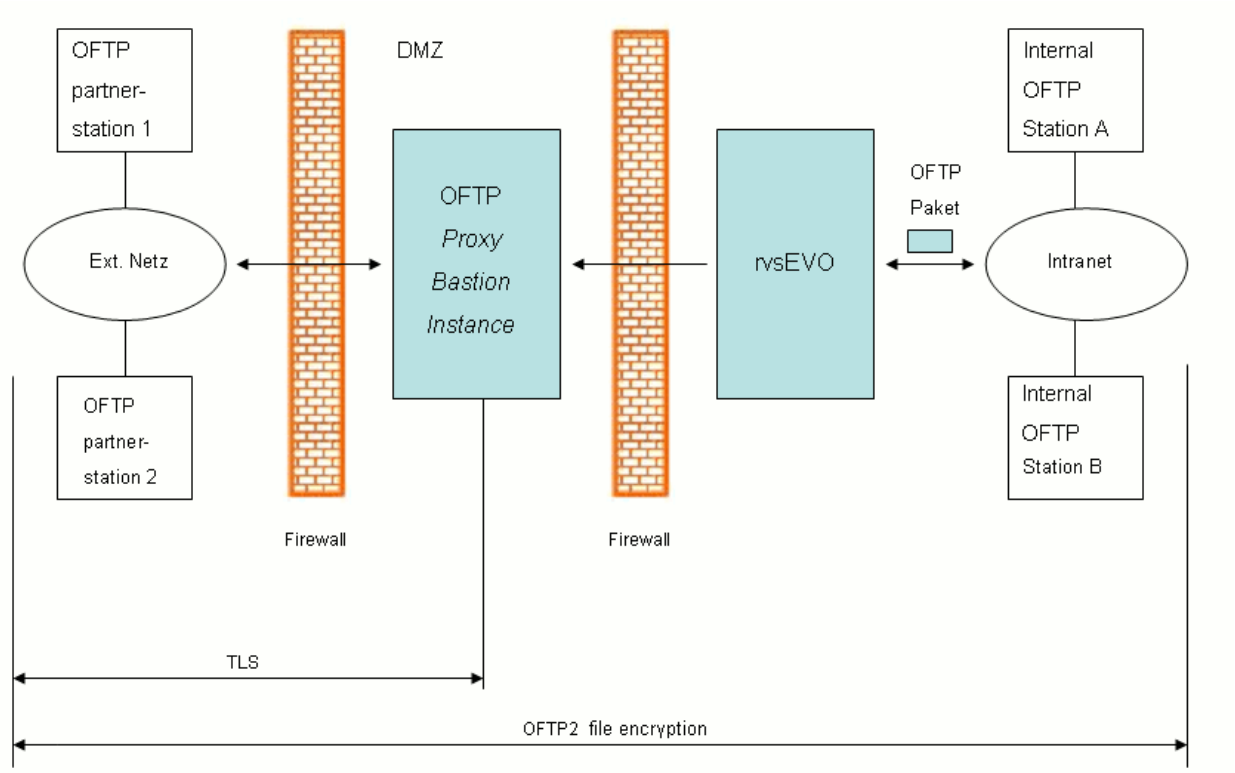
Internet is increasingly used also for transferring the commercial and business-critical data of the content. Such advantages are deciding factors for this, such as its wide distribution, enormous available bandwidths and the reasonable transfer costs.

These advantages are accompanied by the risks to be considered seriously: in internet all data are susceptible to the attacks of their confidentiality and integrity. Their authentication is also not ensured without further ado. This may result into that the data are repudiated by its obvious sender. Such risk is encountered with suitable protocols, such as TLS and HTTPS. For CAD data and also for EDI and any other data, OFTP version 2 has been standardized as a secured internet-protocol.

Any further risk arises through malware, which is spread widely and automatically in internet. Virus, worms, Trojan and other harmful programs can damage the IT-infrastructure of the company considerably. In order to minimize this risk, network areas are developed which are separated from the corporate networks through the firewalls (also referred to as demilitarized Zone - DMZ). For example, http-proxies are set in DMZ, which transport the user data without allowing direct connections between internet and intranet. rvs® OFTP Proxy offers the similar solution, which can be implemented in DMZ in order to transfer the controlled data of OFTP protocol version 1 and 2 between intranet and internet

17.2 rvs® OFTP Proxy Architecture

The rvs® OFTP Proxy is run with two instances: bastion instance and connection instance, which is integrated in rvsEVO. Following figure should illustrate these facts:



The connections are received from the external network using Bastion Instance. These external connections are forwarded to rvsEVO.

The other way round rvsEVO uses the bastion instance for establishing a connection to the OFTP station in the external network.

Communication between rvsEVO and bastion instance is executed exclusively through the connections built by rvsEVO. Communication is restricted to one port, the configurable RMI-port of the bastion instance.

Proxy listener

For being accessible for the external stations one or more Proxy listener are launched by rvsEVO. rvsEVO uses this Proxy listener to establish a connection to the partner station.

Altogether, all functional configurations, keys and certificates required for TLS communication are stored exclusively in rvsEVO.

The Bastion Instance is installed with the program rvs® OFTP Proxy. You can download rvs® OFTP Proxy using the following website:

https://servicenet.t-systems.de/tsi/de/267072/Startseite/Business-Integration/rvs

If you cannot download the software, please contact your sales partner:

phone from Germany: 0800 664 77 45
phone from other countries: +375 606 19 902
E-email: rvs-service@t-systems.com).

We will send you the software also on a DVD.

Carry out the installation of the Bastion Instance, as explained in „User Manual rvs® OFTP Proxy“.

17.3 Bastion Instance

In order to create a connection with the Bastion Instance, it is required to store the parameter in rvsEVO (Bastion Instances) and to configure a proxy listener.

A list of already created Bastion Instances can be displayed in Stations perspective. Please read Chapter 5.6.2 "Display Proxy Bastions" for more information.

Add a Bastion

You can add a Bastion Instance via Stations perspective. Please read Chapter 5.6.1 "Add Proxy Bastion" to find out how to add a Bastion.

Configure Proxy listener

A receiver/listener must be defined for each Bastion Instance. For this purpose, please proceed, as described in chapter 5.2.1 "Setting up of a Local Station"

You also have to add a neighbour station with Proxy TCP/IP or Proxy TLS network, for the communication via rvs® OFTP Proxy. Please read chapter 5.3.2 "Setting up of a Neighbour Station" for detailed information.

First the Bastion Instance should be started and following the rvsEVO server.

Hints:

Failure of connection attempts

You can start a process after connection problems with rvs® OFTP Proxy. This functionality is to be set via ODETTE parameters **ConnSetupProxyFailCount** and **ConnSetupProxyFailScript** (see chapter 10.7 "ODETTE Parameters").

Keep connection

You can configure a time interval for sending heartbeats to prevent the interrupt of connection if there is no transmission. This functionality is to be set up in parameter `listener.controller-thread.sleep-time` and `listener.controller-thread.keep-alive-counter` in `$RVSPROXY_HOME\conf\RemoteListenerProperties.properties` file of rvs® OFTP Proxy program.

Use the `listener.controller-thread.sleep-time` parameter to set up the sleep time (in milliseconds) of the controller thread before it checks next time for status of listener.

Use the `listener.controller-thread.keep-alive-counter` parameter to define the number of checkups before the next heartbeat is sent.

18 File Service Module

18.1 Basics

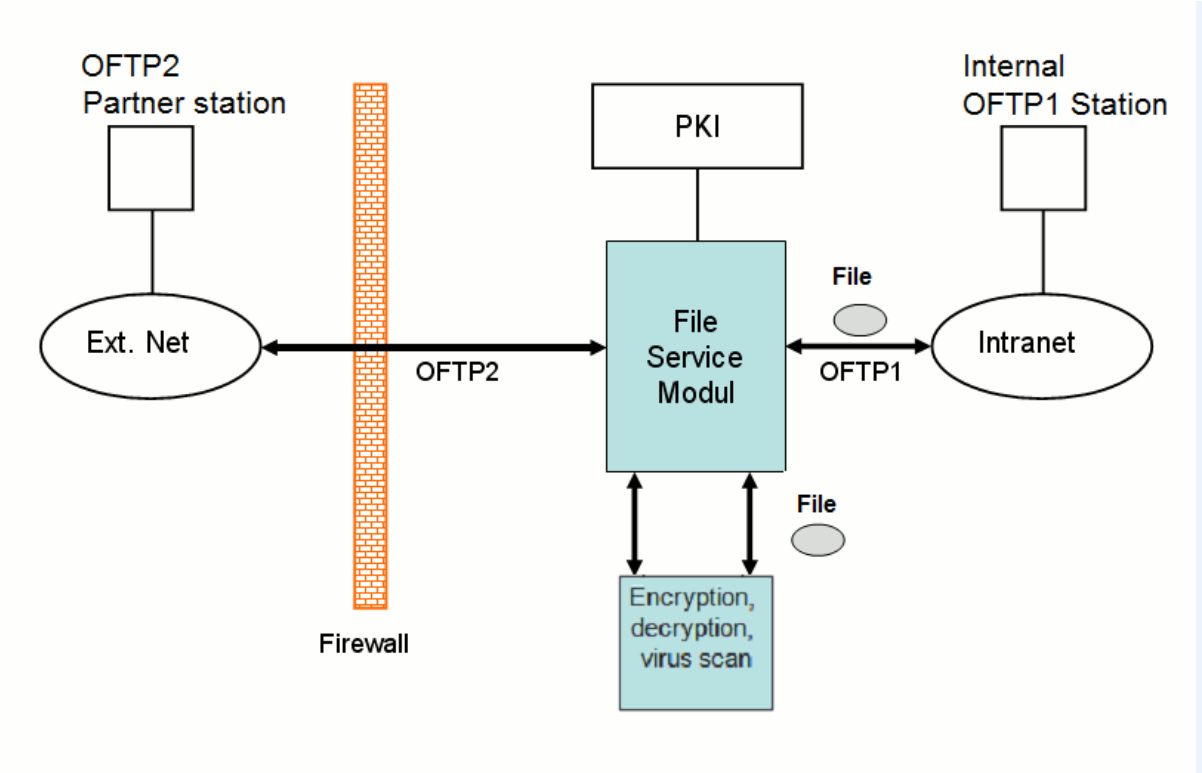
The internet is increasingly used for exchanging commercial and business critical data, which are especially security-relevant. OFTP version 2 is standardized as a secure internet protocol for CAD data, EDI and any other data.

File Service Module works as an OFTP router which de- and encrypts files accordingly OFTP2 for stations which do not support OFTP2 itself.

18.2 Architecture of File Service Module

The File Service Module works as a OFTP router and de- and encrypt files for internal stations substitutional before he forwards them.

The illustration below shows the File Service Module architecture:



Hint: You can combine the File Service Module with rvs® OFTP Proxy (see chapter 17 "rvs® OFTP Proxy").

The following functions are available:

– The File Service Module works as an OFTP router

– Protocol alternation OFTP1 <-> OFTP2, all elements, EERP, even

– En- and decryption of files acting for stations

– Automatc Certificate Exchange acting for stations

– Virus scan of received files
– Connection to the PKI for downloads of certificates, OCSP check, download of CRL

18.3 Setting up of a Neighbour Station for File Service Module

In this chapter is described the adjustment of setting up of a neighbour station for the OFTP File Service Module. Please read chapter 5 "Management of Stations" for detailed description of station configuration.

The relevant parameter for File Service Module is:

Required File Service Module parameter

Parameter	Description
File Service Module	This parameter enables the File Service Module. Possible values: - NONE: File Service Module is not needed (default) - INTERNAL: this partner station is an internal station (located in the intranet). When receiving an encrypted file which is routed to an internal station the mechanism of encryption at the File Service Module is automatically activated, depending on the encryption component used by the sending partner station. - EXTERNAL: this partner station is marked as an external station (located in the internet); for this option all encryption and compression features are available.

The next parameters are only available if FileProxy Service = EXTERN. The table “Security parameters:” on page 84 shows a description of the parameters.

– Security Feature Set
– Offline Compression
– Encryption
– File Signature
– Requested signed EERP/NERP
– Encryption Algorithm

Hint: Both, direct neighbour stations and routing stations can be marked by File Service Module. Therewith the configured File Service Module function is station wide fixed and can not be overwritten while creating a transmission. This corresponds with a forced encryption during activated encryption (Security=FORCED).

19 rvsEVO Database

From rvsEVO version 6.00 upwards the following data are written into the database:

– Jobdata

– Data of user management

– Data of rvsEVO configuration

– Data of key management

– Data of jobstarts

– Data of Observer.

The following databases are available: **Derby** for Standard und Tiny Edition, **Microsoft SQL** and **Oracle** for Enterprise Edition. While installation of rvsEVO you have to define what database is to be installed. (See chapter 2.3 "New Installation of rvsEVO")

19.1 Derby

Derby database is a free, java based relational database from Apache Foundation. If you select Derby Embedded during the installation procedure, a database will be automatically (by rvsEVO) installed in the directory `$RVS_HOME/db`.

19.2 Microsoft SQL Server

Pre-conditions

The following pre-conditions must be fulfilled if you decided for an MS SQL database:

– a MS SQL user was set up with server role PUBLIC and CONNECT permission.

– a MS SQL database was created and the user was defined as database owner.

– rvsEVO was tested with MS SQLServer 2008 R2.

MS SQL connection parameters

During rvsEVO installation process (see chapter 2.4 "Fresh installation of rvsEVO") the following MS SQL connection parameters must be set:

– `jdbc.url`, `jdbc.user`, `jdbc.password`

– `jdbc.url` has the following syntax:

```
jdbc:sqlserver://<host>:<port>;database=<name>
```

Example:

```
jdbc:sqlserver://10.33.72.18:1433;databaseName
=rvsdb
```

Default port for MS SQL is 1433.

19.3 Oracle

Pre-conditions

The following pre-conditions must be fulfilled if you decided yourself for an Oracle database:

– an Oracle database user was set up with the user rights `connect`, `resource`, `create session` and `create table`.

– your Oracle configuration is accurate. If an Oracle client is installed on your machine, you can test the configuration with the following command:

```
sqlplus ORACLE-user@ORACLE-Network servername/
password
```

Example:

```
sqlplus skk@RVS.TSYSTEMS.DE/skk
```

If this command could be executed successfilly (if a user exists and a database is available), you can start the rvsEVO installation.

The following Oracle database was tested:

– Oracle 11g

Connection parameters

While installation of rvsEVO (see Chapter 2.4 "Fresh installation of rvsEVO") the following Oracle connection parameters have to be set:

`jdbc.url`, `jdbc.user`, `jdbc.password` und `Net Service Name`.

`jdbc.url` has the following Syntax:

```
jdbc:oracle:thin:@<server>:<port>/<service_name>
```

Example:

```
jdbc:oracle:thin:@localhost:1521/rvsORA
```

Default port for Oracle is `1521`.

Note: If Oracle is running on an external computer, this computer must be defined in the Oracle configuration file:

```
$ORACLEHOME/network/admin/tnsnames.ora
```

Example (tnsnames.ora):

```
#TNSNAMES.ORA Network Configuration File: /opt/oracle/
product/8.1.7/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.


RVS.TSYSTEMS.DE =

  (DESCRIPTION =

    (ADDRESS_LIST =

      (ADDRESS = (PROTOCOL = TCP)(HOST = rvsaix3)(PORT
= 1521))
```

```
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rvsDB)
    )
  )
```

In this example `RVS.TSYSTEMS.DE` is an Oracle network service name and `rvs` a service name. A network service name is needed to identify an Oracle database in a network. This name should not be mixed up with the global database name, although it has a similar syntax. The service name identifies a database instance, because several instances can run on the same computer.

`jdbc.user` is the Oracle user and `jdbc.passsword` is his password.

Note: The Oracle connection data from the installation window are stored in the file `$RVS_HOME/conf/jdbc.properties`.

19.4 Connection Pooling

With MS SQL- or Oracle database rvsEVO offers the functionality **connection pooling**. That means there is a pool which contains connections to the database. If a connection is needed, one connection of the pool can be used and it is not necessary to establish a connection for every access. rvsEVO uses `C3P0` (Hibernate) for connection pooling.

The settings for Connection Pooling can be done via `$RVS_HOME/conf/jdbc.properties` file.

Example of configuration with default values:

```
c3p0.minPoolSize=5
c3p0.maxPoolSize=25
c3p0.checkoutTimeout=0
c3p0.maxIdleTime=0
c3p0.idleConnectionTestPeriod=60
c3p0.automaticTestTable=POOLING_TEST
c3p0.acquireRetryAttempts=30
c3p0.acquireRetryDelay=1000
c3p0.breakAfterAcquireFailure=false
```

The most important parameter is `c3p0.idleConnectionTestPeriod`. With this parameter you define the time interval (in seconds) for considering the connections of the pool. The parameter `maxPoolSize` defines the maximum number of connections to the database and `minPoolSize` defines the initial number of connections.

For further information please read the C3P0 documentation:

http://community.jboss.org/wiki/HowToconfiguretheC3P0connectionpool

19.5 How to Delete and Create the Database Tables

Hints:

– The scipts for deletion and creating database tables are located in $RVS_HOME/system/db directory.

– Stop rvsEVO before executing the database scripts and restart rvsEVO after creating the new tables.

Remove tables

If the data are not required anymore the current database tables can be droped via the script $RVS_HOME/system/db/drop.

Create tables

After the deletion the database tables must be created again.

Use the following scripts for the creation of the database tables:

– $RVS_HOME/system/db/create for the rvsEVO user and configuration table.

– $RVS_HOME/system/db/create_default_user for creation of the default user.

– $RVS_HOME/system/db/init_config for the configuration table with default values and the values changed during installation

– $RVS_HOME/system/db/init_identity for the assingment of the rights to the user roles

19.6 Database Maintenance

Below you can find a description of the scripts for importing and exporting data into / from the database. The scripts are located in $RVS_HOME/tools directory.

Attention: With Derby database, the rvsEVO server must be stopped before running the skripts. With Oralce and MS SQL database the scripts can also be executed with a started server. But we highly recommend to stop the rvsEVO server before launching the input scripts.

Exeption: the downloadCRLs tool must be lauched with started rvsEvo server.

Hint: The following error messages indicate that another process has access to the database. You have to stop this process before starting the script again.

– java.sql.SQLException: Failed to start database 'C:/rvsEVO/db' with class loader sun.misc.Launcher...

– java.sql.SQLException: Another instance of Derby may have already booted the database C:\rvsEVO\db.

19.6.1 createDefaultQueries

Use this tool for creating default queries for Monitor Log messages (see chapter 7.4 "Monitor Log Queries") and job lists (see chapter 6.5 "Display of Receive and Send Jobs").

You only can execute this tool successfully if there are no default queries in the database anymore (queries called ACTIVE SEND JOBS, ACTIVE RECEIVE JOBS, ENDED JOBS, FAILED JOBS or MONITOR_LOG_QUERY). Follow these steps to establish wether there are queries, named as mentioned before and to rename the queries if necessary.

- Export the existing queries with `exportQueries` script into a xml file (see chapter 19.6.20).
- Parameter <name> refers to the query name. Edit this parameter if necessary.
- Import the queries from xml file into the database with `importQueries` script and delete the existing queries with `-clear` parameter (see chapter 19.6.21).

If there are no queries with default query name in the database you can execute `createDefaultQueries` script.

Syntax:
`createDefaultQueries`

19.6.2 downloadCRLs

Use this tool for importing CRL's which are needed for certificates with CRL validation into the database. rvsEVO server must be started.

Syntax:
`downloadCRLs [-help] [-?] [-verbose|-v]`

Optional Parameters

Parameter	Description
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.3 exportCertificateUsageDefinition: Export of the Definition of Certificate Usage

Use this tool to export the definition of certificate usage from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportCertificateUsageDefinition -f <File> [-r] [-
help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the data are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.4 importCertificateUsageDefinition: Import of the Definition of Certificate Usage

Use this tool to import the definition of certificate usage into the database and delete the present database entries.

Syntax:

```
importCertificateUsageDeninition -f <File> [-c] [-
help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	to delete the old data before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.5 exportConfiguration: Export of rvsEVO Configuration Data

Use this tool to export the rvsEVO configuration data from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportConfiguration -f <File> [-r] [-help] [-?] [-
verbose|-v]
```

Required Parameter

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the configuration data are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.6 importConfiguration: Import of rvsEVO Configuration Data

Use this tool to import the rvsEVO configuration data into the database and delete the present configuration data.

Syntax:
`importConfiguration -f <File> [-c] [-help] [-?] [-verbose|-v]`

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	to delete the old data before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.7 exportCRLs: export of a CRL

Use this tool to export a CRL (**C**ertificate **R**evocation **L**ist) from the database into a file and replace a file with the same name if necessary.

Syntax:
`exportCRLs -f <File> [-r] [-help] [-?][-verbose|-v]`

Erforderlicher Parameter:

Parameter	Beschreibung
-f <File> **-file <File>**	Path and name of the file in which the CRL is to be imported.

Optionale Parameter:

Parameter	Beschreibung
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.8 exportJobs: Export of all Send and Receive Jobs

Use this tool to export the transmission jobs from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportJobs -f <File> [-r] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the transmission jobs are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.

Optional Parameters:

Parameter	Description
-v **-verbose**	Verbose message output.

19.6.9 importJobs: Import of Send and Receive Jobs

Use this tool to import the transmission jobs into the database and delete the present transmission data.

Syntax:

```
importJobs -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old data before new transmission data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.10 exportJobstarters: Export of Jobstart Files

Use this tool to export the configured jobstarts from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportJobstarters -f <File> [-r] [-help] [-?]
[-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the jobstart data are to be imported

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.11 importJobstarters: Import of Jobstart Files

Use this tool to import the data of jobstarts into the database and delete the present jobstart data.

Syntax:

```
importJobstarters -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old data before new jobstart data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.12 exportKeys: Export of Keys and Certificates

Use this tool to export the certificates from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportKeys -f <File> [-r] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the certificates are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.13 importKeys: Import of Keys and Certificates

Use this tool for importing keys and certificates into the database and deleting the present key data.

Syntax:

```
importKeys -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old data before new keys and certificates data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.14 importKeystores: Import of a Keystore File

Use this tool to import the certificates of a keystore file into the database. This functionality is important if your rvsEVO is updated from version 5.xx to 6.00 or upwards.

Syntax:

```
importKeystores [-help] [-?] [-verbose|-v]
```

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old certificates before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.15 importLicense: Import of rvsEVO License Key

Use this tool to import a new license key at runtime. The old license file is renamed in „license.properties.“ plus date pattern and the new one is saved in $RVS_HOME\ conf directory and renamed in „license.properties“. A Monitor-Log message gives you information about the success of the operation:

LICENSE_IMPORTED: successful import

LICENSE_NOT_IMPORTED: import was not successful

Syntax:

```
importLicense -f <File> [-verbose] [-help] [-?]
```

Required Parameter:

Parameter	Description
-f <File>	Path and name of the license file.

Optional Parameters:

Parameter	Description
-help	Displays a description of the current command.
-?	Displays a description of the current command.
-verbose	Verbose message output.

19.6.16 exportObservers: Export of the Configured Observers

Use this tool to export the Observer data from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportObserver -f <File> [-r] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the Observer data are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.17 importObservers: Import Observer Data

Use this tool for importing Observer data into the database and deleting the present Observers.

Syntax:

```
importObserver -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old Observer data before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.18 exportOftpProxies: Export of Proxy Bastion Instances

Use this tool to export the created Proxy Bastion Instances from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportOftpProxies -f <File> [-r] [-help] [-?] [-
verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the Proxy Bastion Instances are to be imported

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.19 importOftpProxies: Import of Proxy Bastion Instances

Use this tool for importing Proxy Bastion Instances into the database and deleting the present key data.

Syntax:

```
importOftpProxies -f <File> [-c] [-help] [-?] [-
verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	To delete the old Proxy Bastions before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.20 exportQueries: Export of Queries

Use this tool to export the queries from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportQueries -f <File> [-r] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the queries are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.21 importQueries: Import of Queries

Use this tool for importing the previously exported queries into the database and deleting the present queries.

Syntax:

```
importQueries -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	to delete the old queries before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.22 exportStationlist: Export of the Stationlist

Use this tool to export the stationlist from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportStationlist -f <File> [-remote] [-r|-replace] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File>	Path and name of the file in which the stationlist is to be imported.

Optional Parameters:

Parameter	Description
-remote	This parameter availbles to start the tool during rvsEVO is running.
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.23 importStationlist: Import of a Stationlist

Use this tool for importing a stationlist into the database and deleting the present station settings.

Syntax:

```
importStationlist -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	to delete the old station data before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.24 importTrustedServiceList: Import of a TSL

Use this tool to import a TSL (**T**rust **S**ervice Status **L**ist) into the rvsEVO database.

Syntax:

```
importTrustedServiceList -f <TSL filename> -u <URI>
[-remote] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <TSL filename>	Path of the file which includes the TSL **Hint**: Only one parameter can be set: -f or -u
-u <URL>	URL for CRL download **Hint**: Only one parameter can be set: -f or -u

Optional Parameters:

Parameter	Description
-remote	Availbles to start the tool during rvsEVO is running.
-verbose	Verbose message output.
-help	Displays a description of the current command.
-?	Requests help information.

19.6.25 exportUsers: Export of User Data

Use this tool to export the user data from the database into a file and replace a file with the same name if necessary.

Syntax:

```
exportUsers -f <file> [-r] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file in which the user data are to be imported.

Optional Parameters:

Parameter	Description
-r **-replace**	To replace a file with the same name if necessary. If a file with the same name is present and this parameter was not set the process is failed with an error message.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.26 importUsers: Import of User Data

Use this tool for importing user data into the database and deleting the present users.

Syntax:

```
importUsers -f <File> [-c] [-help] [-?] [-verbose|-v]
```

Required Parameter:

Parameter	Description
-f <File> **-file <File>**	Path and name of the file from which the data are to be imported into the database.

Optional Parameters:

Parameter	Description
-c **-clean**	to delete the old user data before new data are imported.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.27 switchDatabase: Exchange of Database

Use this tool to exchange the rvsEVO database.

Syntax:

```
switchDatabase -dbtype <type> [-driver <driver> -url
<url> -username <name> -password <password> -jdbcfile
<path> -skiptest -switchonly -testonly] [-help|-h] [-
?] [-verbose|-v]
```

Required Parameters:

Parameter	Description
-dbtype	Type of database. Possible values: - derby - oracle - mssql

Optional Parameters:

Parameter	Description
-driver	Name and path of jdbc driver
-url	jdbc URL; see chapter Kapiteln 19.2 (MS SQL) und 19.3 (Oracle)
-username	Login name of database user
-password	Password of database user
-jdbcfile	If the parameters are imported via imputfile: path and name of the file
-skiptest	Skip test connection to database
-switchonly	Switch database without export and import of data.
-testonly	Make only test connection to database by using new parameters.
-h **-help**	Displays a description of the current command.
-?	Displays a description of the current command.
-v **-verbose**	Verbose message output.

19.6.28 Command Line Tools for Backup and Recovery

The description of the following programs, you can find in chapter `21 "Backing Up and Recovering rvsEVO Data"`

`backup:` Export of the whole database

`restore:` Import of the whole database

19.6.29 Command line tools for internal use

The following programs are only for internal use:

- `importCRLs`

20 Archiving the rvsEVO Data

The archiving process takes place automatically after transmission jobs are terminated or failed. During the process the job data are written into the RevisionLog file in the `$RVS_HOME\archive` directory by default. Name and path of this file can be set up in **PersistenceArchive** parameter. (Please see chapter 10.5 "MAINTENANCE"). The specified name is extended by a timestamp and a sequential counter initiating with „0000“. (Format: `Filename.xml.yyMMddHHmmss_cccc`).

The following settings of RevisionLog file can be done via parameter group `MAINTENANCE` of `Parameter` perspective:

The maximum size and maximum number of revision files can be indicated via **MaxRevisionLogSize** and **MaxRevisionLogCount** parameters. If the value in **MaxRevisionLogSize** parameter is reached a new RevisionLog file is created.

Deletion of job data from database

Any terminated and failed jobs older than 7 days will be deleted every 24 hours, by default. This configuration can be set up via the parameters **CleanupDays**, **CleanupInterval** and **CleanupTime** (please see chapter 10.5 "MAINTENANCE" for getting the description of the parameters).

Hint: You can also start the deletion of job data by launching the script `archiveJobs` at the command prompt as an alternative. (See chapter 12.8 "archiveJobs: Deletion of Job Data from Database").

Deletion of an Individual Job

You also have the option to delete an individual job. This can be reached via the script `archiveJob` (see chapter 12.9 "archiveJob: Deletion of an Individual Job from the Database") or via the GUI (see chapter 6.6 "Operations on Jobs").

21 Backing Up and Recovering rvsEVO Data

You can back up all relevant rvsEVO data and recover them if necessary. This is particularly important when an error has occurred in rvsEVO and the user wishes to revert to the old status prior to the error.

21.1 Backup

Pre-condition

The rvsEVO server has to be stopped before the backup function can be launched.

Start the `$RVS_HOME/tools/backup` script to perform the backup functionality.

Syntax:
`backup [-u] [-s] [-verbose][-help][-?]`

All parameters are optional.

Backup Parameters:

Parameter	Description
-file <file> **-f**	Without function
-userfiles **-u**	Use this parameter to save the user data from `Files` directory (subdirectories `inbox`, `outbox` and `temp`)
-skipjobs **-s**	Use this parameter for desabling backup of job-data.
-?	Requests help information.
-help **-h**	Displays a description of the current command.
-verbose	Verbose message output

By default, the data are saved in `$RVS_HOME/archive` directory. Every backup process creates a new subdirectory. The directory name constists of „EVO6B-“, date and a Time. (Format: `EVO6B-YYYYMMTT-hhmmss`).

What is backed up?

The following directories are backed up:

- `$RVS_HOME/conf`
- `$RVS_HOME/files`
- `$RVS_HOME/system`
- `$RVS_HOME/jobs` (if used)

The following data from the database are backed up:

– configuration data
– stationlist
– OFTP-Proxy Bastions
– jobdata
– jobstarts
– certificates and their assignments
– Observer
– user data

21.2 Recovering the rvsEVO Data

Use the `$RVS_HOME/tools/restore.bat` script to recover the rvsEVO data.

Syntax:
`restore [-b <file>] [-t <file>] [-u] [-s] [-i] [-verbose] [-help][-?]`

All parameters are optional.

Restore Parameters:

Parameter	Description
-backup <file> **-b**	Name of the backup file
-temp **-t**	Directory for temporary data
-userfiles **-u**	Parameter for importing the user data (see also the same parameter of backup script)
-skipjobs **-s**	Parameter for disabling the import of job data.
-initdb **-i**	Parameter for initializing the database before restore.
-?	Requests help information.
-help **-h**	Displays a description of the current command.
-verbose	Verbose message output

22 PKI Connection

In this chapter the configuration and the functionality of the rvsEVO PKI binding will be described. You also get information about certificate validation via GUI.

22.1 Introduction

In a PKI (**P**ublic **K**ey **I**nfrastructure) public keys with their certificates are stored.

Following features are available for rvsEVO PKI binding:

– obtaining of public keys and certificates from a PKI for a partner station.

– certificate validation with OCSP (**O**nline **C**ertificate **S**tatus **P**rotocol), CRL (**C**ertificate **R**evocation **L**ist) or CertPath.

– Transfer of information from a OCSP server by HTTP

Hint: From rvsEVO version 6.00 upwards certificate validation via OCSP and CRL is also available for TLS certificates.

22.2 Configuration of PKI Binding

The PKI binding should be configured in the `Stations` perspective via the OFTP parameters **Certificate Validation Type** and **PKI**.

Hint: Usually the PKI binding should be configured for a partner station. For the local station it can be interesting, only if you want to validate your own certificate.

Parameters for PKI Binding

Parameter	Description
Certificate Validation Type	This parameter describes the type of the certificate validation. The following three types are supported: - `OCSP` (Online Certificate Status Protocol) - `CRL` (Certificate Revocation List) - `CertPath`. `None` in this field means: No validation will be done.
PKI	If the checkbox is activated the PKI functionality is used.

22.3 Validation Methods

Which validation method will be used depends on security standards in the company.

OCSP If you use OCSP as validation method, the OCSP service must be available for the rvsEVO nodes.

CRL The validation with CRL can be done offline. The rvsEVO user/administrator can provide the list of certificates (CRL) and store it manually or download the list via the key management of rvsEVO (see 22.4 "Certificate Validation via Key Management Perspective"). The disadvantage of this way is, that certificate can expire between two updates.

CertPath For CertPath validation you only need the certificate to be validated and the certificates of the issuer up to the root certificate. Normaly these certificates change rarely. So not much service is necessary. With CertPath validation the formal correctness of the certificate, the certificate of the issuer ... up to the root certificate is checked. It is not checked whether a certificate was revocated.

Default Procedure with Encryption via TLS

You need to set up parameter **Certificate Validation Type** to `CRL` or `OCSP` for CRL or OCSP validation. Else CertPath validation takes place. With all outgoing calls certificate validation takes place. To validate certificates of incoming calls parameter **Client Authentification** is to be set up to `NEEDED` or `WANTED` (see table “TLS and Proxy TLS Parameters (Local)” on page 49).

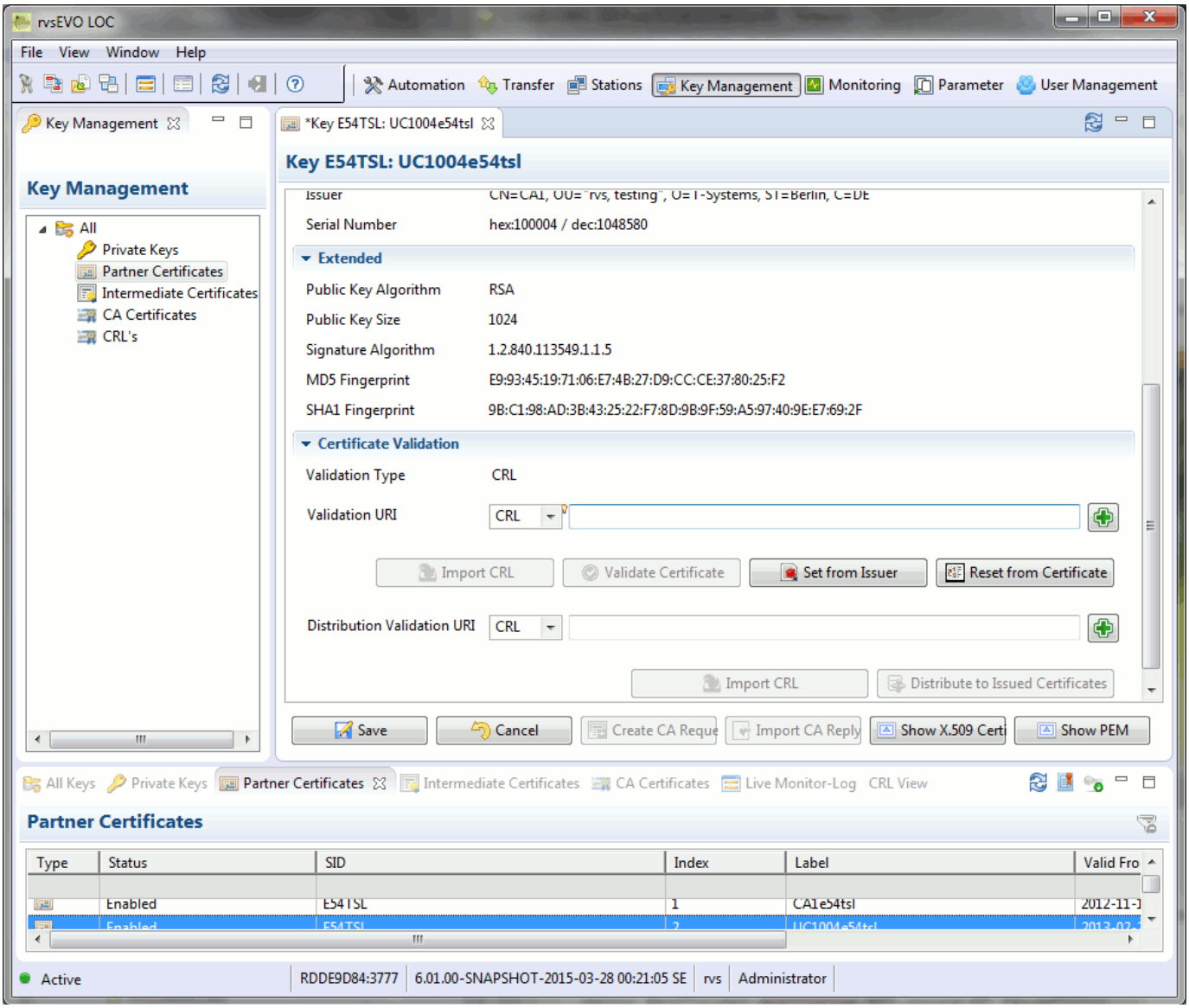
Default Procedure with File Encryption

There is no certificate validation by default.Validation with OCSP or CRL always includes the validation of the chain. For using OCSP and CRL validation you need to set up parameter **Certificate Validation Type**.

Hint: rvsEVO don’t accepets expired certificates.

22.4 Certificate Validation via Key Management Perspective

A certificate can also be validated via Key Management perspective in the detailed view of the key, section `Certificate validation`. The detailed view can be opened by a double-click on the certificate. Precondition is that the parameter **Certificate Validation Type** was set up (see chapter 22.2).



Validate certificate

To validate a certificate:

- Specify the OCSP address or download address of CRL in input field **Validation URI** and select the validation type (OCSP or CRL) from the combo box. Press to specify more than one address and to open a further input field for validation URI or press to delete an entry.

  Hint: Via mouse click on **Reset from Certificate** button the validation URI is read from the certificate.

By using CA certificates you can transmit the validation URI of the distributor, which is specified in input field **Distribution Validation URI** to all his certificates. This functionalty is executable in the detailed view of a CA certificate via a mouse click on **Distribute to Issued Certificates** button. In the detaild view of a partner certificate or an intermediate certificate you have the possibility to read the validation URI via mouse click on the button **Set from Issuer**.

- Press the **Validate Certificate** button to validate the certificate.

Import / update CRL

Hint: A current CRL can be downloaded by pressing the **Import CRL** button. The CRL is imported into the rvsEVO system. Set the security parameter **CRLUseHttpProxy**=Y for downloading the CRL via Http Proxy server. Via security parameter **CRLAutoDownload** you can configure that the CRL is downloaded and imported during the next establishment of connection.

Via security parameters **CRLAutoUpdate** and **CRLAutoUpdateInterval** you can configure that automatic updates should take place (See chapter 10.9 "Security Parameters" for the description of the parameters).